

Article

Energy-Aware Persistent Multi-UAV Coverage via Reinforcement Learning Guided by User Priority and Outage

Haoyu Mei , Chengtao Xu *, Ruozhe Li  and Xueshan Luo

National Key Laboratory of Information Systems Engineering, National University of Defense Technology, Changsha 410073, China; meihao20@nudt.edu.cn (H.M.); liruo2he@nudt.edu.cn (R.L.); xsluo@nudt.edu.cn (X.L.)

* Correspondence: xuchengtao@nudt.edu.cn

Highlights

What are the main findings?

- POGS-QMIX improves geometric service availability for high-priority users across the evaluated operating conditions; the reported multi-seed results quantify both central tendency and variability.
- The coupled priority–outage coordination interface and return-energy mask provide structured guidance for assignment, low-level control, and energy-aware action selection.

What are the implications of the main finding?

- Jointly considering priority and consecutive-outage history offers a practical way to trade high-priority continuity against aggregate service and lower-priority intra-class fairness.
- Following configuration-specific training, the framework retains high-priority service continuity across variations in fleet size, user population, coverage radius, and UAV speed, while energy-stress tests quantify coverage and return failures under reduced battery capacity and charging rate constraints.

Abstract

In disaster response and other infrastructure-limited settings, UAV-mounted access points can rapidly restore service availability for mobile ground users as demand and fleet availability evolve. Existing single-slot coverage formulations, however, can mask prolonged individual outages and do not jointly represent heterogeneous service priorities, finite battery capacities, and periodic recharging. We study persistent geometric multi-UAV service coverage, where a user is available for service when it lies inside a UAV footprint. We propose Priority- and Outage-Guided Safe QMIX (POGS-QMIX), a hybrid hierarchical framework in which a centralized online coordinator forms conflict-reduced UAV–user targets from fleet-wide priority and outage information, while parameter-shared QMIX agents independently choose target-conditioned low-level actions. The framework couples class-balanced outage memory, assignment, dense target-progress feedback, and a return-energy action mask. The evaluation includes learning and non-learning baselines, greedy-versus-Hungarian assignment, multi-seed statistics, sensitivity studies, operating-condition studies, and energy-stress tests. In the default scenario, POGS-QMIX obtains high-priority coverage 0.547 ± 0.009 and maximum high-priority outage 38.0 ± 4.7 slots over five independent seeds.

Keywords: multi-UAV coverage; multi-agent reinforcement learning; QMIX; persistent coverage; consecutive outage; return-energy management



Academic Editor: Ya Zhang

Received: 6 August 2026

Revised: 7 September 2026

Accepted: 8 September 2026

Published: 10 September 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](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Reliable communication is among the first services required after earthquakes, floods, wildfires, and other large-scale emergencies. Terrestrial base stations, backhaul links, and power supplies may be damaged precisely when first responders and affected populations need voice, video, positioning, and situational-awareness data. Restoration of fixed infrastructure can be slow. By contrast, UAV-mounted base stations or relays can be deployed rapidly, repositioned as conditions evolve, and operated through favorable air-to-ground links [1–3].

Keeping such an aerial network operational is a sequential service problem rather than a one-time deployment. As evacuation routes, search-and-rescue activity, and temporary-shelter locations change, so does the spatial distribution of communication demand. Users such as first responders can require higher service priority than routine users. Meanwhile, every UAV has finite battery energy and must periodically leave the service region to recharge; when it does, the remaining UAVs must redistribute service and later coordinate its reintegration. Persistent operation therefore requires coordinated coverage handovers and charging rotations that avoid prolonged disconnections, especially for urgent users.

Traditional UAV coverage studies commonly optimize instantaneous spatial coverage or finite-horizon traversal of a region [4–6]. These objectives are valuable for deployment and path planning, but a high spatial or time-averaged coverage score can still conceal long service gaps for individual mobile users. Persistent coverage consequently requires joint decisions on user selection, motion, coverage handover, and energy replenishment over a long horizon [7,8].

Two properties make this problem fundamentally different from conventional area coverage. First, instantaneous or aggregate coverage can conceal severe temporal inequality. A policy may repeatedly serve easily reached users while leaving a smaller group continuously disconnected. This effect is particularly consequential when users have heterogeneous priorities. Second, coverage, motion, and charging decisions are coupled across UAVs: each UAV observes only part of the system, while its action changes fleet-level service and future energy availability. Cooperative multi-agent reinforcement learning (MARL) is therefore suitable because it can learn adaptive decentralized policies from shared team outcomes with mobile-user dynamics. However, an unconstrained end-to-end MARL policy must simultaneously discover spatial division of labor, long-range navigation, and return-and-charging behavior. Sparse service feedback and a large joint action space make effective exploration difficult, while a penalty-only energy model leaves return-risky actions available during exploration.

Within this cooperative MARL setting, QMIX provides the low-level backbone for the discrete-action fleet: centralized training uses the global state for team credit assignment, while monotonic value factorization preserves consistency between joint-value maximization and independently greedy low-level actions [9]. During online operation, POGS-QMIX combines centralized fleet-wide target coordination with decentralized target-conditioned low-level control. It uses current geometric coverage, consecutive interruption duration, energy consumption, and estimated return-energy status as service information. Its contributions are summarized as follows.

- We formulate priority-aware persistent geometric service coverage by jointly retaining mobile-user priority, user-level consecutive-outage memory, finite normalized battery budgets, and periodic recharging. The formulation explicitly separates service amount, interruption tails, intra-class fairness, energy use, and return failures.
- The main algorithmic contribution is a coupled online interface that maps class-balanced priority and outage state, geometry, and energy risk into conflict-reduced targets, then exposes those targets consistently to agent observations, transition-level

progress feedback, and feasible-action selection. This closed-loop coupling provides the task-specific coordination structure for persistent service.

- We define a model-based return-energy mask used consistently in behavior selection and TD target construction, relate reserve selection to a generic sufficient-bound template for explicit abstract bounds, and evaluate reserve sensitivity empirically.
- We compare against value-decomposition, policy-gradient, random, and assignment-guided non-learning controllers; compare greedy and Hungarian matching; report detailed implementation and evaluation settings; and quantify independent-seed variability using sample standard deviations and confidence intervals.

The remainder of this paper is organized as follows. Section 2 reviews persistent coverage, energy-aware operation, and cooperative MARL. Section 3 develops the system model and mathematical problem formulation. Section 4 presents POGS-QMIX, and Section 5 reports the experimental results. Section 6 concludes the paper and outlines future work.

2. Related Work

2.1. From Complete Coverage to Persistent Coverage

Persistent surveillance concerns repeated service over an indefinite horizon, for which revisit intervals, information age, and worst-case service gaps are central performance measures [7]. By contrast, finite-horizon coverage path planning seeks to traverse a region once while reducing omissions, repeated paths, and collisions. Representative approaches range from geometric and heuristic methods to online cooperative learning [4–6]. A single-pass coverage ratio therefore does not characterize persistent service. It can remain high even when a subset of locations experiences long periods without service.

Early persistent-coverage studies primarily used combinatorial optimization and receding-horizon planning. Continuous coverage with charging constraints can be cast as a partitioning problem over energy-constrained tours originating from charging stations [8]. Persistent surveillance under communication constraints has been formulated to minimize revisit idle time subject to bounded data-delivery latency [10], while mobile-charger planning combines area partitioning, cyclic routes, and replenishment schedules to control long-term regional age [11]. These formulations offer explicit feasibility or approximation properties, but their static points of interest, known task graphs, or deterministic visit times do not directly represent randomly moving users with time-varying service urgency.

2.2. Long-Term Service Quality and User Fairness

Learning-based policies adapt to unknown dynamics through interaction. A recurrent actor–critic method with wonderful-life utility minimizes maximum regional age for periodic coverage [12], whereas distributed deep reinforcement learning (DRL) jointly optimizes time-average communication coverage, geographic fairness, and flight energy [13]. Age-of-information (AoI)-oriented sweep coverage explicitly relates data freshness to continuous dwell time at points of interest [14], and dynamic-demand planning couples user visits with charging-station stops [15]. Recent communication-coverage studies also optimize long-term throughput, user association, and Jain fairness [16] or jointly account for throughput, fairness, and inter-UAV connectivity [17].

These metrics characterize complementary aspects of long-term service. Regional age is appropriate for periodic visitation, but it does not directly measure the fraction of time during which a mobile user receives communication service. Aggregate coverage and throughput can favor users that are easier to reach. The Jain index measures allocation balance, but a high value can coexist with uniformly poor service and does not constrain the

tail of consecutive interruptions. For heterogeneous-priority users, class-specific coverage, intra-class fairness, and user-level outage duration should therefore be evaluated jointly.

2.3. Energy Constraints, Return-Energy Management, and Recharging Scheduling

Long-duration operation makes energy feasibility a condition for mission continuity rather than merely a path cost. Multi-robot rendezvous planning uses integer programming and graph transformations to schedule periodic meetings between charging robots and task UAVs [18]. Rotational recharge scheduling minimizes the fleet size required to sustain service at heterogeneous locations [19]. For fixed-area coverage, combining path allocation, optimal flight speed, detailed energy estimation, and battery constraints reduces flight energy [20].

Under stochastic mobility and fleet state conditions, exact combinatorial control rapidly becomes intractable. Multi-agent double deep Q-networks (DDQNs) and transfer learning can jointly learn data collection, collision avoidance, and return-to-charge behavior [21]; optimization-assisted DRL further balances service level, fairness, deployment energy efficiency, and low-battery risk [22]. Nevertheless, encoding residual energy only in the state or reward leaves return-risky actions available during exploration. This observation motivates an action-level filter based on the estimated return energy from each candidate next position.

2.4. Multi-Agent Reinforcement Learning and Cooperative Structures

For multi-UAV coverage, structured representations and inductive biases can improve learning efficiency. Multimodal image–vector states encode dynamic user distributions, coverage regions, and UAV operating modes [23]; trajectory pheromones reduce redundant coverage through indirect coordination [24]. Transformers with parameter sharing support variable numbers of UAVs and users [25], while symmetry-preserving graph networks exploit rotational equivariance in large-scale continuous control [26].

At the cooperative-learning level, Value-Decomposition Networks (VDNs) represent the team value as the sum of individual values [27], whereas QMIX uses a state-conditioned monotonic mixing network to improve representational capacity while preserving decentralized greedy action consistency [9]. Centralized-critic methods provide an alternative for policy optimization; Multi-Agent Proximal Policy Optimization (MAPPO) has demonstrated strong performance on cooperative benchmarks [28]. These backbones address multi-agent credit assignment, but persistent UAV coverage additionally requires cross-slot service memory, spatial division of labor, and explicit return-energy constraints.

Table 1 positions POGS-QMIX relative to persistent and energy-aware approaches. Its focus is the control problem of coupling heterogeneous-priority outage memory to assignment-conditioned MARL and an online return-energy filter.

Table 1. Scope comparison with representative long-term and energy-aware coverage methods. A check mark means that the feature is explicit in the cited formulation.

Method	Mobile Demand	User-Level Outage Memory	Heterogeneous Priority	Periodic Charging	Online Assignment + MARL
Liu et al. [13]	✓	–	–	–	–
Sun et al. [12]	–	regional age	–	–	–
Wang et al. [15]	✓	–	–	✓	–
Cheng et al. [16]	✓	–	–	limited-energy return	–
Proposed POGS-QMIX	✓	✓	✓	✓	✓

POGS-QMIX combines user-level consecutive-outage priority with a deterministic one-to-one coordinator, target-conditioned value factorization, and a consistent model-based return-energy mask in exploration and bootstrapping. The contribution lies in this closed-loop system design for persistent priority-aware service.

3. System Model and Problem Formulation

We consider a cooperative UAV fleet that provides persistent coverage to mobile ground users with heterogeneous priorities. The fleet operates over a finite mission horizon with mobile users, limited battery capacity, and periodic recharging. Performance is characterized by class-specific coverage, intra-class fairness, consecutive-outage statistics, priority-weighted energy efficiency, and failed-return events.

Figure 1 provides an intuitive overview of the considered scenario. Multiple UAVs coordinate their coverage of spatially distributed mobile users from heterogeneous priority classes, while their residual battery levels and distances to charging stations constrain continued service and may trigger return-to-charge maneuvers. Accordingly, the problem is to learn a joint UAV policy that allocates service targets, motion, and charging opportunities over time so as to sustain differentiated coverage, limit long user outages, and manage return risk under the modeled energy constraints.

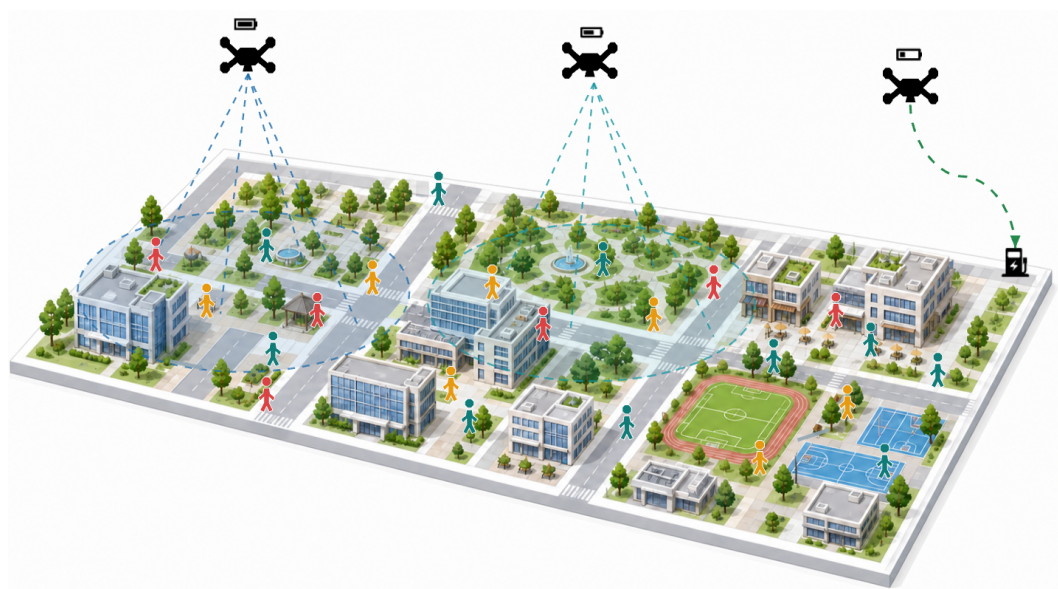


Figure 1. Illustrative persistent multi-UAV coverage scenario. User colors distinguish heterogeneous priority classes, dashed regions depict UAV coverage footprints, and the routed UAV illustrates a battery-driven return-to-charge maneuver.

3.1. Network, User, and Coverage Models

Consider the rectangular mission region

$$\Omega = [0, L_x] \times [0, L_y], \quad (1)$$

where L_x and L_y are its side lengths. The system contains N UAVs and M ground users, indexed by

$$\mathcal{N} = \{1, \dots, N\}, \quad \mathcal{M} = \{1, \dots, M\}. \quad (2)$$

Time is slotted as $t \in \{0, 1, \dots, T-1\}$, where T denotes the mission horizon. The positions of UAV i and user j at the beginning of slot t are

$$\mathbf{p}_i(t) = [x_i(t), y_i(t)]^T, \quad \mathbf{u}_j(t) = [x_j(t), y_j(t)]^T. \quad (3)$$

The residual battery energy of UAV i is $b_i(t) \in [0, B_{\max}]$. The region contains K fixed charging stations at

$$\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_K\}. \quad (4)$$

A UAV replenishes its battery after entering a station neighborhood and suspends coverage while charging. Persistent operation must therefore coordinate service delivery with return and replenishment decisions.

Each slot follows a fixed causal order. The state at the beginning of slot t determines target assignments, observations, and action masks. The UAVs then execute their actions and reach $\mathbf{p}_i(t+1)$. Coverage $C_j(t)$ and energy consumption $e_i(t)$ are evaluated using the post-action UAV positions and the current user positions $\mathbf{u}_j(t)$; battery and consecutive-outage states are updated accordingly. Finally, users move to $\mathbf{u}_j(t+1)$. Thus, every decision at slot t depends only on information available before action execution.

3.1.1. User Priorities and Service Values

Each user j has a fixed priority label

$$\ell_j \in \{\text{H}, \text{M}, \text{L}\}, \quad (5)$$

for high-, medium-, and low-priority service, respectively. Let w_{ℓ_j} be the service-value weight associated with label ℓ_j , and define

$$w_j = w_{\ell_j}, \quad w_{\text{H}} > w_{\text{M}} > w_{\text{L}} > 0. \quad (6)$$

These weights encode relative service importance; their numerical values are specified in Section 5.

3.1.2. User Mobility Model

Ground users follow a bounded isotropic random walk. At every slot and for every user, independently sample an angle $\Theta_j(t) \sim \mathcal{U}[0, 2\pi)$ and a normalized step magnitude $R_j(t) \sim \mathcal{U}[0, 1]$, and set $\boldsymbol{\zeta}_j(t) = R_j(t)[\cos \Theta_j(t), \sin \Theta_j(t)]^T$. Thus, step magnitude—rather than area density—is uniform, and $\|\boldsymbol{\zeta}_j(t)\|_2 \leq 1$. Let v_g denote the maximum displacement per slot. User j updates its position according to

$$\mathbf{u}_j(t+1) = \Pi_{\Omega}(\mathbf{u}_j(t) + v_g \boldsymbol{\zeta}_j(t)), \quad (7)$$

where $\Pi_{\Omega}(\cdot)$ denotes Euclidean projection onto Ω . Random generators for initialization, mobility, exploration, replay sampling, and network initialization are seeded separately from a recorded master seed. The control formulation itself depends only on observed user positions and remains compatible with other mobility processes.

3.1.3. UAV Motion and Action Space

At each slot, UAV i selects an action from

$$a_i(t) \in \mathcal{A} = \{0, 1, \dots, 9\}, \quad (8)$$

where action 0 denotes hovering, actions 1–8 denote the eight cardinal and diagonal movement directions, and action 9 invokes return to the nearest charging station. Let $\mathbf{d}(a)$

be the corresponding direction vector, with $\mathbf{d}(0) = \mathbf{0}$ and unit norm for actions 1–8, and let v_u be the UAV displacement per slot. For $a_i(t) \neq 9$,

$$\mathbf{p}_i(t+1) = \Pi_{\Omega}(\mathbf{p}_i(t) + v_u \mathbf{d}(a_i(t))). \quad (9)$$

For the return action, UAV i selects its nearest charging station,

$$\mathbf{c}_i^*(t) = \arg \min_{\mathbf{c}_k \in \mathcal{C}} \|\mathbf{p}_i(t) - \mathbf{c}_k\|_2, \quad (10)$$

Let $D_i^c(t) = \|\mathbf{c}_i^*(t) - \mathbf{p}_i(t)\|_2$. The UAV travels the distance

$$\Delta_i^{\text{ret}}(t) = \min\{v_u, D_i^c(t)\}, \quad (11)$$

along the station direction:

$$\mathbf{p}_i(t+1) = \Pi_{\Omega}\left(\mathbf{p}_i(t) + \Delta_i^{\text{ret}}(t) \frac{\mathbf{c}_i^*(t) - \mathbf{p}_i(t)}{\max\{D_i^c(t), \epsilon\}}\right), \quad (12)$$

where $\epsilon > 0$ prevents division by zero. The station-neighborhood radius is denoted by r_c .

Charging is governed by an explicit state transition rather than an unlisted eleventh action. Let $\mathcal{A}_{\text{dep}} = \{1, \dots, 8\}$ denote departure actions, and let $D_i^c(t+1) = \min_k \|\mathbf{p}_i(t+1) - \mathbf{c}_k\|_2$. Action 0 means “continue charging” when $z_i(t) = 1$, while action 9 means “enter/continue return” and docks the UAV when the station neighborhood is reached. The charging indicator is

$$z_i(t+1) = \begin{cases} 1, & z_i(t) = 1 \text{ and } a_i(t) = 0, \\ 1, & D_i^c(t+1) \leq r_c \text{ and } a_i(t) = 9, \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

When $z_i(t) = 1$, action 0 holds the position fixed and applies the charge update; a feasible action in $\mathcal{A}_{\text{dep}}$ sets $z_i(t+1) = 0$ and uses the normal motion and discharge equations. A fully charged UAV may still wait by selecting action 0. This rule makes docking, continued charging, and departure reproducible without adding a separate charging action.

3.1.4. Coverage Model

We adopt a geometric service-availability model. Let R_c be the coverage radius, and let $z_i(t+1) \in \{0, 1\}$ indicate whether UAV i is charging after action execution. Pairwise coverage is

$$C_{ij}(t) = \mathbb{I}(\|\mathbf{p}_i(t+1) - \mathbf{u}_j(t)\|_2 \leq R_c, z_i(t+1) = 0). \quad (14)$$

User j is covered if at least one non-charging UAV serves it:

$$C_j(t) = \max_{i \in \mathcal{N}} C_{ij}(t). \quad (15)$$

Hence, $C_j(t) = 1$ denotes geometric service availability in slot t , whereas $C_j(t) = 0$ denotes a geometric outage. This geometric abstraction evaluates footprint-level service availability independently of altitude-dependent path loss, fading, SINR, inter-cell interference, scheduling, association capacity, and data-rate sharing. Equation (15) permits any number of users inside a footprint to be counted simultaneously.

3.2. Persistent Service State and Energy Constraints

To distinguish a transient interruption from a prolonged service gap, we maintain the consecutive-outage length $\tau_j(t)$:

$$\tau_j(t+1) = \begin{cases} 0, & C_j(t) = 1, \\ \min\{\tau_j(t) + 1, T_{\text{clip}}\}, & C_j(t) = 0, \end{cases} \quad (16)$$

where T_{clip} is the clipping and normalization horizon; in the experiments, $T_{\text{clip}} = T$. Define the normalized outage state

$$\bar{\tau}_j(t) = \frac{\tau_j(t)}{T_{\text{clip}}} \in [0, 1]. \quad (17)$$

The normalized state permits outage costs to remain comparable across mission horizons, while a convex function of $\bar{\tau}_j(t)$ assigns progressively greater importance to longer interruptions.

Energy Consumption and Recharging Model

The energy model uses normalized energy units (EU) and comprises movement energy, a baseline airborne-operation cost, and a service-load proxy. One slot is the simulation control interval; B_{max} and b_i are measured in EU, r_b in EU/slot, η_m in EU/m, η_h in EU/slot, and η_c in EU/(served-user slot). Let

$$d_i(t) = \|\mathbf{p}_i(t+1) - \mathbf{p}_i(t)\|_2 \quad (18)$$

denote the traveled distance, and let $n_i^{\text{cov}}(t)$ be the number of users covered by UAV i . For a non-charging UAV,

$$e_i(t) = \eta_m d_i(t) + \eta_h + \eta_c n_i^{\text{cov}}(t), \quad (19)$$

where η_m , η_h , and η_c are the movement, baseline-operation, and service-load coefficients, respectively. The battery evolves as

$$b_i(t+1) = \max\{0, b_i(t) - e_i(t)\}. \quad (20)$$

For a charging UAV, service is suspended and battery energy increases at rate r_b :

$$b_i(t+1) = \min\{B_{\text{max}}, b_i(t) + r_b\}. \quad (21)$$

The estimated movement energy required to reach the nearest charging station from position $\mathbf{p}$ is

$$\hat{E}^{\text{ret}}(\mathbf{p}) = \eta_m \min_{\mathbf{c}_k \in \mathcal{C}} \|\mathbf{p} - \mathbf{c}_k\|_2. \quad (22)$$

The movement-only estimate defines the nominal model-based return boundary. The reserve s_b covers baseline airborne operation, service load, and model uncertainty during the return. Let L_{max} denote an assumed upper bound on the number of slots required to reach a charging station, $n_{\text{max}}^{\text{cov}}$ an assumed upper bound on the number of users simultaneously served by a returning UAV in one slot, and δ_e an assumed upper bound on unmodeled one-slot energy error. Under these assumptions, $s_b \geq L_{\text{max}}(\eta_h + \eta_c n_{\text{max}}^{\text{cov}} + \delta_e)$ is a generic sufficient-bound template for reserve design. The experiments select s_b empirically; Section 5.6 reports $s_b = 25$ EU as the smallest tested reserve associated with zero observed failed-return events across the evaluated scenarios.

3.3. Performance Metrics and Optimization Objective

Policy performance is evaluated with class-specific service, temporal continuity, fairness, energy efficiency, and return-failure metrics.

Let $\mathcal{M}_\ell$ denote the set of users in priority class $\ell \in \{H, M, L\}$. Its coverage score is

$$S_\ell = \frac{1}{|\mathcal{M}_\ell|T} \sum_{j \in \mathcal{M}_\ell} \sum_{t=0}^{T-1} C_j(t). \quad (23)$$

S_ℓ is a dimensionless fraction in $[0, 1]$: the mean fraction of the mission for which users in class ℓ receive geometric service.

Define the user-level coverage score

$$s_j = \frac{1}{T} \sum_{t=0}^{T-1} C_j(t). \quad (24)$$

The Jain fairness index within priority class ℓ is

$$J_\ell = \frac{\left(\sum_{j \in \mathcal{M}_\ell} s_j\right)^2}{|\mathcal{M}_\ell| \sum_{j \in \mathcal{M}_\ell} s_j^2 + \epsilon}. \quad (25)$$

The Jain index is dimensionless and lies near $[0, 1]$; a value close to one indicates an even distribution of service within class ℓ , but does not imply a high service amount.

Temporal continuity is measured by

$$O_{\text{mean}} = \frac{1}{MT} \sum_{t=0}^{T-1} \sum_{j=1}^M \tau_j(t), \quad O_{\text{max}} = \max_{j,t} \tau_j(t), \quad (26)$$

together with $O_H^{\text{max}} = \max_{j \in \mathcal{M}_{H,t}} \tau_j(t)$. Both outage quantities are measured in slots. These statistics distinguish policies with similar time-average coverage but different interruption tails.

Priority-weighted energy efficiency (PWEE) measures delivered priority-weighted covered-user slots per normalized energy unit:

$$\text{PWEE} = \frac{\sum_{t=0}^{T-1} \sum_{j=1}^M w_j C_j(t)}{\sum_{t=0}^{T-1} \sum_{i=1}^N e_i(t) + \epsilon}. \quad (27)$$

Its unit is weighted covered-user slots/EU. The numerator is not additionally normalized or scaled, so PWEE comparisons use the same priority weights and EU convention for every method. Finally, a failed-return event occurs when a UAV battery transitions from a positive value to zero outside every charging-station neighborhood. Counting only the positive-to-zero transition makes this an event count rather than a per-slot failure duration:

$$F_{\text{ret}} = \sum_{t=0}^{T-1} \sum_{i=1}^N \mathbb{I}[b_i(t) > 0, b_i(t+1) = 0] \\ \times \mathbb{I}[\|\mathbf{p}_i(t+1) - \mathbf{c}_i^*(t+1)\|_2 > r_c]. \quad (28)$$

Together, these metrics separate service amount, temporal continuity, intra-class allocation, energy use, and observed return failures.

Optimization Objective

Subject to the motion, coverage, battery, charging, and estimated return-energy models above, the goal is to learn a joint policy π that maximizes differentiated persistent service

while controlling consecutive outages, energy expenditure, and failed returns. Section 4.6 constructs a compact team reward from normalized service objectives; the metrics in Section 3.3 remain independent evaluation criteria. The learning objective is

$$\max_{\pi} J(\pi) = \mathbb{E}_{\pi} \left[\sum_{t=0}^{T-1} \gamma^t r(t) \right]. \quad (29)$$

where $\gamma \in (0, 1]$ is the discount factor and the expectation is over mobility, environment transitions, and policy actions. Dynamic users, partial observability, combinatorial joint actions, and long-horizon energy coupling motivate a cooperative MARL solution.

4. Priority- and Outage-Guided Safe QMIX

POGS-QMIX uses QMIX as its low-level learning backbone because persistent fleet coverage is a fully cooperative problem with discrete per-UAV actions and a combinatorial joint action space. The state-conditioned monotonic mixer uses global information during centralized training to assign credit across agents while preserving consistency between joint-value maximization and independently greedy low-level actions [9]. During online operation, the target coordinator uses a shared fleet-and-user snapshot to construct fleet-wide assignments, after which each UAV selects its target-conditioned low-level action independently. Accordingly, the complete architecture combines centralized training, online coordination, and decentralized low-level control.

To convert this cooperative backbone into a persistent-coverage controller, POGS-QMIX organizes two decision levels over three coupled temporal horizons. At the long-horizon level, cross-slot memory records each user's priority, current service state, and consecutive outage. At the coordination level, an online coordinator combines this service state with current fleet geometry and energy risk to assign a distinct target to each available UAV. At the immediate-control level, parameter-shared individual Q-networks select target-conditioned motion actions, and a model-based mask removes actions that violate the estimated return-energy boundary. During learning, the QMIX mixer combines the selected individual values using the global state.

Figure 2 summarizes the resulting data flow. The upper path operates during both training and online control. Persistent-service states generate conflict-reduced targets through the coordinator, and these targets condition the shared low-level action-value network. Each UAV then applies its local return-energy mask and selects an action independently. The lower path operates only during centralized training. Complete episodes, including the action-availability masks, are sampled to construct masked Double-Q targets and update the shared agent network and QMIX mixer. The deterministic, state-dependent coordinator structures the service targets, while the learned low-level controller determines UAV motion. The remainder of this section follows this input–output chain. Section 4.2 defines the Dec-POMDP and value-factorization backbone; Sections 4.3 and 4.4 map priorities, outage histories, fleet states, and user geometry into target-conditioned agent information; Section 4.5 maps candidate moves and battery states into action availability; Section 4.6 maps post-action service and energy outcomes into the team reward; and Sections 4.7 and 4.8 provide the training update and complexity analysis.

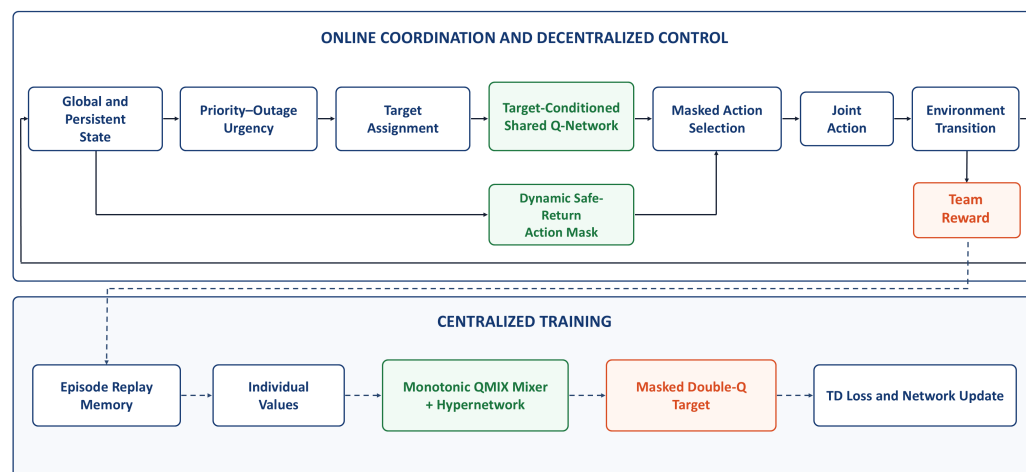


Figure 2. Architecture of POGS-QMIX with centralized training, online coordination, and decentralized low-level control. The online coordinator receives a fleet-and-user snapshot, constructs priority–outage targets, and broadcasts the compact snapshot and target identities. Each UAV then evaluates its own target-conditioned action values, applies its local return-energy mask, and selects a low-level action independently. During training only, selected individual values enter the monotonic QMIX mixer, and masked Double-Q targets provide the TD update.

4.1. Design Rationale: Coupled Decision Horizons

POGS-QMIX couples three decision horizons rather than treating its components as independent auxiliary modules. These horizons characterize the temporal role of information in the control loop rather than asynchronous update frequencies, and all three are evaluated at each slot. At the long-horizon service-memory level, the normalized consecutive-outage state $\bar{\tau}_j(t)$ records which users have remained without service and the relative duration of each interruption. Class-balanced priority weights preserve service differentiation without allowing the more numerous low-priority class to dominate the fleet-level signal. This level identifies the users that require intervention.

At the coordination level, the pair score $A_{ij}(t)$ converts user urgency and UAV–user geometry into a one-to-one target allocation. The resulting target $j_i^*(t)$ determines the service responsibility of each UAV, reduces duplicated pursuit, and converts the global heterogeneous-service requirement into a compact agent-specific objective. At the immediate control level, the target-conditioned value $Q_i(o_i, a_i)$ determines the motion of the assigned UAV, while the availability mask excludes actions that violate the estimated model-based return-energy boundary.

The three levels form the closed decision loop

$$\begin{aligned}
 \underbrace{\{\bar{\tau}_j(t), \tilde{w}_j\}_{j=1}^M}_{\text{service memory}} &\longrightarrow \underbrace{\{A_{ij}(t), j_i^*(t)\}}_{\text{fleet coordination}} \\
 &\longrightarrow \underbrace{\{Q_i(o_i, a_i), m_i(a_i, t)\}}_{\text{masked action selection}}.
 \end{aligned} \tag{30}$$

The executed actions determine coverage and energy consumption, which in turn update the service memory for the next slot. Consequently, service differentiation and temporal continuity shape coordination, coordination structures decentralized credit assignment, and the return-energy layer filters actions near the estimated return boundary. This coupling is the central design principle of POGS-QMIX.

4.2. Dec-POMDP and Value Factorization

The low-level cooperative-control problem is modeled as a decentralized partially observable Markov decision process (Dec-POMDP)

$$\mathcal{G} = \langle \mathcal{N}, \mathcal{S}, \{\mathcal{A}_i\}, P, \{\mathcal{O}_i\}, O, r, \gamma \rangle. \quad (31)$$

The global state concatenates, in ascending index order, five normalized scalars per UAV (two-dimensional position, battery, charging indicator, and return indicator), seven scalars per user (position, one-hot priority, current coverage, and normalized outage), two coordinates per station, and normalized mission time. Its dimension is $5N + 7M + 2K + 1$ (456 in the default setting).

For each UAV, the observation concatenates: (i) nine own-state scalars (position, battery, charging/return indicators, vector and distance to the nearest station, and time); (ii) five scalars for every other UAV, ordered by UAV index; (iii) $K_u = 3$ urgent-user slots from each priority class; and (iv) one assigned-target slot. Each user slot has nine scalars: relative position (2), normalized distance, class-balanced weight, linear and squared outage, coverage indicator, assignment score, and a validity bit. Users are sorted within each class by decreasing urgency $q_j(t)$, with distance and user index as deterministic tiebreakers. Missing slots are zero padded with validity zero. The assigned-target slot follows the same layout. Thus, the observation dimension is $9 + 5(N - 1) + 27K_u + 9 = 119$ for $N = 5$ and $K_u = 3$. The availability mask is a separate 10-bit vector and is not counted as an observation feature.

Let $Q_i(o_i, a_i; \theta)$ denote the shared individual action-value function evaluated from the current target-conditioned observation o_i . QMIX represents the joint value as

$$Q_{\text{tot}}(\mathbf{o}, \mathbf{a}, \mathbf{s}; \theta, \phi) = f_{\text{mix}}(Q_1, \dots, Q_N, \mathbf{s}; \phi), \quad (32)$$

State-conditioned hypernetworks generate nonnegative mixing weights, enforcing

$$\frac{\partial Q_{\text{tot}}}{\partial Q_i} \geq 0, \quad \forall i. \quad (33)$$

The monotonicity constraint makes decentralized greedy low-level actions consistent with maximization of Q_{tot} for the assigned targets.

Monotonicity constrains the signs of mixer sensitivities but not their magnitudes. A state-conditioned hypernetwork can therefore assign different partial derivatives to the individual Q_i . Priority weights apply to users, while homogeneous UAVs change targets over time. We consequently evaluate class-wise Jain indices as independent service-allocation outcomes alongside coverage and outage.

4.3. Class-Balanced Priority and Outage State

Directly assigning w_ℓ to every user makes the total contribution of a class depend on its population. Let $M_\ell = |\mathcal{M}_\ell|$. We define the class-balanced user weight

$$\tilde{w}_j = \frac{w_{\ell_j}^p / M_{\ell_j}}{\frac{1}{M} \sum_{k=1}^M w_{\ell_k}^p / M_{\ell_k}}, \quad (34)$$

where $p \geq 1$ controls priority separation. The denominator normalizes the mean user weight to one, so inter-class preference is decoupled from class population. For assignment, define $\hat{w}_j = \tilde{w}_j / \max_k \tilde{w}_k \in (0, 1]$.

Using (17), define the service urgency of user j as

$$q_j(t) = \lambda_p \hat{w}_j \left[1 + \lambda_o \bar{\tau}_j^2(t) + \lambda_c (1 - C_j(t)) \right], \quad (35)$$

where $\lambda_p, \lambda_o, \lambda_c \geq 0$. The quadratic term increases marginal urgency with interruption duration, while $1 - C_j(t)$ distinguishes currently uncovered users. Consequently, among users of similar priority, sustained interruption receives progressively stronger assignment pressure.

4.4. Priority–Outage-Guided Target Assignment

For an available UAV i and user j , let the normalized travel distance be

$$\bar{d}_{ij}(t) = \frac{\|\mathbf{p}_i(t) - \mathbf{u}_j(t)\|_2}{D_{\max}}, \quad (36)$$

where $D_{\max} = \sqrt{L_x^2 + L_y^2}$. The return-energy risk of UAV i is

$$\rho_i^E(t) = \min \left\{ 2, \frac{\hat{E}^{\text{ret}}(\mathbf{p}_i(t))}{\max\{b_i(t), \epsilon\}} \right\}. \quad (37)$$

The assignment score is

$$A_{ij}(t) = q_j(t) - \lambda_d \bar{d}_{ij}(t) - \lambda_e \rho_i^E(t), \quad (38)$$

where λ_d and λ_e balance service urgency against travel and return risk. Candidate pairs are sorted by $A_{ij}(t)$ and greedily accepted subject to one target per UAV and at most one assigned UAV per user. Charging UAVs are excluded from matching. The resulting $O(NM \log(NM))$ procedure produces conflict-reduced targets without altering the low-level action space.

Each accepted target $j_i^*(t)$ is held fixed for the ensuing transition. Its nine-dimensional feature is the user-slot layout specified above, including a validity bit; a zero vector denotes the absence of a valid target. The same deterministic ordering and padding rules are used during training and evaluation.

4.5. Dynamic Return-Energy Action Mask

For candidate action a , let $\tilde{\mathbf{p}}_i(t+1, a)$ and $\tilde{b}_i(t+1, a)$ denote the predicted post-action position and battery after movement and baseline airborne consumption; an action that keeps the UAV at a charging station instead includes the charging increment. For every non-return action, infeasibility is detected by

$$\tilde{b}_i(t+1, a) < \hat{E}^{\text{ret}}(\tilde{\mathbf{p}}_i(t+1, a)) + s_b, \quad (39)$$

where s_b is an empirical reserve intended to absorb baseline, service-load, and model-error terms. The availability mask is

$$m_i(a, t) = \begin{cases} 1, & a = a_{\text{ret}}, \\ 0, & a \neq a_{\text{ret}} \text{ and (39) holds,} \\ 1, & \text{otherwise.} \end{cases} \quad (40)$$

The return action a_{ret} is always retained to avoid an empty feasible set. If the current battery satisfies $b_i(t) \leq \hat{E}^{\text{ret}}(\mathbf{p}_i(t)) + s_b$ outside a station neighborhood, all other actions are masked. Charging and departure follow (13).

Applying (40) at every decision step enforces the nominal movement-energy boundary during behavior exploration and TD-target maximization. The reserve provides an empirical margin for baseline operation, service load, and model uncertainty; the reported failure counts and sensitivity results quantify its behavior with the tested model settings.

4.6. Compact Persistent-Service Team Reward

The shared reward contains five scale-controlled components with distinct roles. The class-balanced instantaneous service utility is

$$U_{\text{cov}}(t) = \frac{\sum_{j=1}^M \tilde{w}_j C_j(t)}{\sum_{j=1}^M \tilde{w}_j}. \quad (41)$$

The post-service consecutive-outage cost is

$$L_{\text{out}}(t) = \frac{\sum_{j=1}^M \tilde{w}_j \tau_j^2(t+1)}{\sum_{j=1}^M \tilde{w}_j}. \quad (42)$$

Squaring the normalized outage state concentrates the penalty on long interruptions without introducing multiple correlated deficit terms.

Target progress is represented by a potential-difference heuristic. Let $\delta_i(t)$ indicate that UAV i has a valid target, and define the weighted target-distance score

$$\Phi(\mathbf{s}_t) = - \frac{\sum_{i=1}^N \delta_i(t) \tilde{w}_{j_i^*(t)} \|\mathbf{p}_i(t) - \mathbf{u}_{j_i^*(t)}(t)\|_2}{D_{\text{max}} \sum_{i=1}^N \delta_i(t) \tilde{w}_{j_i^*(t)} + \epsilon}, \quad (43)$$

The progress reward is

$$R_{\text{prog}}(t) = \gamma_p \Phi(\mathbf{s}_{t+1}^-) - \Phi(\mathbf{s}_t), \quad (44)$$

where $\gamma_p \in (0, 1]$ is the progress discount and $\mathbf{s}_{t+1}^-$ is the post-action, pre-mobility state used for slot- t service evaluation. Target identities and weights remain fixed within this transition, preventing within-slot switching from creating artificial progress. Because the coordinator can choose a different target at the next slot, (44) is treated as an explicit target-progress reward rather than a stationary policy-invariant PBRS potential. Its effect is isolated by ablation.

Normalized energy consumption is

$$L_{\text{energy}}(t) = \frac{\sum_{i=1}^N e_i(t)}{E_{\text{norm}}}, \quad (45)$$

where E_{norm} is a fixed upper scale for one-slot fleet energy. Finally, $N_{\text{fail}}(t)$ counts newly occurring failed-return events in slot t . The team reward is

$$r(t) = \alpha U_{\text{cov}}(t) - \beta L_{\text{out}}(t) + \chi R_{\text{prog}}(t) - \eta L_{\text{energy}}(t) - \kappa N_{\text{fail}}(t), \quad (46)$$

where all coefficients are nonnegative. The continuous terms are normalized, while a return failure is represented by a one-time event count; all coefficients are fixed before policy evaluation.

Alignment with the Persistent-Coverage Objective

The reward in (46) follows the operational decomposition

$$\begin{aligned} \text{persistent service} &= \text{service amount} \\ &+ \text{temporal continuity} \\ &+ \text{operational sustainability.} \end{aligned} \quad (47)$$

The coverage utility U_{cov} supplies the service-amount signal, whereas the convex outage cost L_{out} supplies a smooth continuity signal that emphasizes users near the tail of the outage distribution. In particular, let $\bar{\tau}_{\max}(t+1) = \max_j \bar{\tau}_j(t+1)$ and $\tilde{w}_{\min} = \min_j \tilde{w}_j > 0$. Equation (42) implies

$$\bar{\tau}_{\max}(t+1) \leq \sqrt{\frac{\sum_j \tilde{w}_j}{\tilde{w}_{\min}}} L_{\text{out}}(t), \quad (48)$$

which links minimization of the differentiable quadratic cost to suppression of the outage upper envelope. Larger priority weights further amplify the contribution of long high-priority interruptions.

The target-progress term provides dense directional feedback between sparse service events by rewarding motion toward an urgent assigned target before it enters the coverage disk. Because it can alter the learned optimum, its coefficient is treated as a design choice and tested in sensitivity analysis. The energy cost discourages unnecessary motion, the failed-return event cost represents an irreversible loss of service capacity, and the action mask filters exploration and bootstrapping according to the estimated boundary. These components represent service amount, interruption continuity, navigation progress, resource conservation, and estimated execution feasibility.

4.7. QMIX Update and Training Procedure

Complete episodes are stored in replay memory. Let $\bar{\theta}$ and $\bar{\phi}$ denote target-network parameters. Double Q-learning selects feasible next actions with the online individual networks and evaluates them with the target networks:

$$\begin{aligned} y_t &= r(t) + \gamma(1 - z_t)Q_{\text{tot}}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}^*; \bar{\theta}, \bar{\phi}), \\ a_i^*(t+1) &= \arg \max_{a: m_i(a, t+1)=1} Q_i(o_i(t+1), a; \theta), \end{aligned} \quad (49)$$

where z_t is the termination indicator. The training loss is

$$\mathcal{L}_{\text{TD}} = \frac{1}{|\mathcal{B}|} \sum_{(t,e) \in \mathcal{B}} \omega_{t,e} [Q_{\text{tot}}(\mathbf{s}_t^e, \mathbf{a}_t^e) - y_t^e]^2, \quad (50)$$

where $\omega_{t,e}$ masks padded time steps. The same availability mask is used for ϵ -greedy behavior and next-action maximization.

Algorithm 1 gives the complete training procedure. Within a slot, the target identity used by the progress term is fixed before action execution. Coverage and energy are then evaluated at the post-action UAV positions and pre-mobility user positions, after which the service-memory variables and user positions are advanced. The coordinator supplies service targets, and the low-level policies control movement, return, and charging actions.

Algorithm 1 Training Procedure of POGS-QMIX

```

1: Input: environment  $\mathcal{G}$ , online parameters  $\theta, \phi$ , and replay memory  $\mathcal{D}$ ; initialize  $\bar{\theta} \leftarrow \theta$ 
   and  $\bar{\phi} \leftarrow \phi$ .
2: for each episode do
3:   Reset the environment and persistent service states; set trajectory  $\mathcal{T} \leftarrow \emptyset$ .
4:   for  $t = 0, \dots, T - 1$  do
5:     Compute  $\tilde{w}_j$ , urgency  $q_j(t)$ , and pair scores  $A_{ij}(t)$ .
6:     Obtain unique targets  $\{j_i^*(t)\}$  by greedy matching and build target-conditioned
       observations.
7:     for each UAV  $i$  do
8:       Predict candidate next states and construct availability mask  $m_i(\cdot, t)$ .
9:       Select  $a_i(t)$  by masked  $\epsilon$ -greedy maximization of  $Q_i(o_i(t), \cdot; \theta)$ .
10:    end for
11:    Execute  $\mathbf{a}_t$  and obtain post-action coverage  $C_j(t)$  and energy consumption  $e_i(t)$ .
12:    Update batteries, consecutive outages, new return-failure events, and then user
       positions.
13:    Evaluate  $r(t)$  by (46) using the targets fixed at the beginning of the slot.
14:    Observe  $\mathbf{s}_{t+1}, \mathbf{o}_{t+1}, \mathbf{m}_{t+1}, z_t$  and append the transition to  $\mathcal{T}$ .
15:    if  $z_t = 1$  then break.
16:  end for
17:  Store the complete trajectory:  $\mathcal{D} \leftarrow \mathcal{D} \cup \{\mathcal{T}\}$ .
18:  if  $\mathcal{D}$  contains a complete mini-batch then
19:    Sample episodes and select feasible next actions with the online individual
       networks.
20:    Compute  $y_t$  by (49) and  $\mathcal{L}_{TD}$  by (50); update  $\theta, \phi$ .
21:    At the target-update interval, set  $\bar{\theta} \leftarrow \theta$  and  $\bar{\phi} \leftarrow \phi$ .
22:  end if
23: end for
24: Output: trained shared agent network and online coordinator.

```

4.8. Computational Complexity Analysis

Let $A = 10$ denote the number of discrete actions and K the number of charging stations. Computing all assignment scores requires $O(NM)$ operations, and sorting the NM UAV–user pairs for greedy one-to-one matching requires $O(NM \log(NM))$ time and $O(NM)$ auxiliary memory. The fixed-order urgent-user features are constructed by ranking users within each priority class for every UAV. The current sorting implementation requires $O(NM \log M)$ time across the fleet, which is asymptotically absorbed by the assignment term. Because the observation of each UAV contains features of the other UAVs, constructing all observations and evaluating the first layer of the parameter-shared individual network contributes $O(N^2)$ time for fixed hidden widths. The return-energy mask predicts A candidate transitions for each UAV and evaluates the nearest station, requiring $O(NAK)$ time and $O(NA)$ mask storage.

The resulting per-slot online complexity is

$$O(NM \log(NM) + N^2 + NAK), \quad (51)$$

with auxiliary memory of $O(NM + N^2 + NA)$. Target assignment dominates when the user population is larger than the fleet, action set, and charging-station set. In the default setting $(N, M, A, K) = (5, 60, 10, 5)$, the coordinator evaluates 300 UAV–user pairs, while the return-energy layer checks 50 candidate actions against five stations.

The online coordinator is implemented at a ground station or fleet edge server with access to the telemetry/control channel. At the beginning of every slot, UAVs upload position, battery, and mode; the ground service supplies user positions, fixed priority labels, and the maintained coverage/outage state. The coordinator broadcasts the resulting

snapshot and unicast target identifiers; each UAV constructs the same ordered observation and computes its action mask locally from its own battery, position, and the known station map. This division places online coordination at the ground station and decentralized low-level action selection on the UAVs.

For a transparent bandwidth estimate, assume 32-bit position/battery fields, a 16-bit clipped outage counter, compact flags/labels, and byte-aligned records. Each of 60 user records occupies 12 bytes and each of five UAV records 14 bytes, giving 6320 bits per shared snapshot. With a 64-byte header plus one-byte target identifier and a two-byte mask/sequence field per UAV, a broadcast implementation transmits approximately 6952 bits/slot, or 6.95 kbps for a 1 s slot (69.5 kbps for a 0.1 s slot). Repeating the snapshot in five unicasts would require approximately 32 kb/slot. These payload estimates exclude link-layer coding, retransmissions, localization traffic, and encryption overhead. The execution model assumes that snapshot and target packets arrive within one control slot. On a missed coordinator packet, a UAV retains its previous target for one slot but recomputes the return mask locally; longer outages trigger return-only operation.

The monotonic mixer is used only during centralized training and does not contribute to online action-selection complexity. Its global-state dimension is $5N + 7M + 2K + 1$ and therefore grows with the configured problem size. By contrast, fixing K_u makes the individual-network input independent of M , although it grows with N through other-UAV features.

5. Experimental Results and Analysis

5.1. Environment and Implementation Settings

The default scenario was a 1000 m $\times$ 1000 m region containing five UAVs and 60 users. User positions were sampled uniformly and evolved according to (7). Each episode lasted 500 slots. The coverage radius was 150 m, and the UAV and maximum user displacements were 40 and 4 m/slot, respectively. Five charging stations were located at the region center and the four quadrant centers; UAVs began at these stations with full batteries. The high-, medium-, and low-priority class probabilities were 0.2, 0.3, and 0.5, with relative service weights 6.0, 2.5, and 1.0. Table 2 summarizes the default environment and implementation settings.

Table 2. Default experimental environment parameters.

Parameter	Value
Region size	1000 m $\times$ 1000 m
Number of UAVs/users	5/60
Episode length	500
Coverage radius	150 m
UAV/user speed	40/4 m/slot
Battery capacity/charge rate	240 EU/15 EU/slot
Return-energy reserve	25 EU
Charging radius	100 m
Charging stations	5
H/M/L user probabilities	0.2/0.3/0.5
H/M/L priority weights	6.0/2.5/1.0
Move/hover/service-load coeffs.	0.022 EU/m; 0.10 EU/slot; 0.020 EU/(user slot)

The reward coefficients in (46) are $\alpha = 3.0$, $\beta = 2.0$, $\chi = 15.0$, $\eta = 0.12$, and $\kappa = 5.0$, with $\gamma_p = 0.99$. In (35)–(38), $\lambda_p = 1.0$, $\lambda_o = 3.0$, $\lambda_c = 0.5$, $\lambda_d = 0.8$, and $\lambda_e = 0.4$. Priority balancing used $p = 1$. These settings were used throughout the experiments, and their influence on the main service, continuity, and energy metrics is examined in Tables 3 and 4.

Table 3. Sensitivity to outage exponent q and return reserve s_b (mean $\pm$ standard deviation across five seeds). PWEE is in weighted covered-user slots/EU; outage is in slots; failures are events/episode.

Factor	Value	S_H	I_H	$O_H^{\max}$	PWEE	F_{ret}
Outage exponent	$q = 1.0$	0.507 ± 0.019	0.879 ± 0.013	22.9 ± 3.8	11.02 ± 0.27	0.00 ± 0.00
	$q = 1.5$	0.536 ± 0.018	0.878 ± 0.012	28.7 ± 4.1	11.28 ± 0.29	0.00 ± 0.00
	$q = 2.0$	0.547 ± 0.009	0.877 ± 0.012	38.0 ± 4.7	11.45 ± 0.31	0.00 ± 0.00
	$q = 2.5$	0.553 ± 0.020	0.866 ± 0.015	46.8 ± 5.6	11.42 ± 0.34	0.00 ± 0.00
Return reserve	10 EU	0.555 ± 0.019	0.875 ± 0.013	36.4 ± 4.9	11.62 ± 0.33	0.42 ± 0.18
	25 EU	0.547 ± 0.009	0.877 ± 0.012	38.0 ± 4.7	11.45 ± 0.31	0.00 ± 0.00
	40 EU	0.531 ± 0.017	0.873 ± 0.014	42.5 ± 5.1	10.98 ± 0.30	0.00 ± 0.00
	60 EU	0.498 ± 0.021	0.862 ± 0.016	51.7 ± 6.4	10.21 ± 0.37	0.00 ± 0.00

Table 4. One-factor coefficient sensitivity using multipliers of the default value (five-seed means).

Setting	S_H	$O_H^{\max}$ (Slots)	PWEE	F_{ret}
0.5β	0.531	49.6	11.38	0.00
2β	0.544	30.8	11.19	0.00
$0.5\lambda_d$	0.557	41.3	11.27	0.00
$2\lambda_d$	0.516	55.1	10.84	0.00
$0.5\lambda_e$	0.558	36.9	11.51	0.18
$2\lambda_e$	0.522	47.8	10.73	0.00
0.5χ	0.472	93.5	10.41	0.00
2χ	0.551	39.9	11.31	0.00
Default	0.547	38.0	11.45	0.00

The implementation was based on EPyMARL. Each value-based agent used a fully connected 64-unit ReLU layer, a 64-unit GRU, and a linear 10-action head, with parameters shared across UAVs. The QMIX mixer used a 32-dimensional embedding and a two-layer hypernetwork with 64 hidden units. Training used Double Q-learning and RMSProp with learning rate 5×10^{-4} , $\gamma = 0.99$, gradient-norm clipping at 10, a 128-episode replay buffer, 32-episode mini-batches, and a target update every 200 episodes. Rewards were standardized during Q-learning. Exploration decreased linearly from 1.0 to 0.05 over 100,000 environment steps.

All learning methods received 150,000 environment steps and were evaluated greedily every 10,000 steps. PPO advantages were normalized per batch; the value-loss coefficient was 0.5 and the maximum gradient norm was 10. Table 5 reports the method-specific settings used in the experiments. The reported baseline hyperparameters were fixed implementation settings used consistently in all comparisons, rather than independently optimized configurations for each baseline.

Table 5. Baseline-specific implementation and optimization settings.

Method Family	Actor/Agent Network	Critic/Mixer	Optimizer and Learning Rate	PPO/TD Details	Update Schedule
VDN/QMIX	FC(64)–GRU(64)–linear	sum / QMIX(32), hypernet(64)	RMSProp, 5×10^{-4}	Double-Q; clip norm 10	batch 32 episodes; target/200 episodes
IPPO	MLP(64,64), ReLU	local value MLP(64,64)	Adam, 3×10^{-4}	GAE 0.95; clip 0.2; entropy 0.01	10 epochs; 4 minibatches
MAPPO	MLP(64,64), ReLU	central MLP(128,128)	Adam, 3×10^{-4}	GAE 0.95; clip 0.2; entropy 0.01	10 epochs; 4 minibatches

The confirmatory protocol used five independent training seeds for every learning method and ablation. Each final checkpoint was evaluated on 64 episodes generated from

a disjoint list of evaluation seeds; the per-training-seed mean is the statistical unit. Tables report mean $\pm$ sample standard deviation across training seeds. Figures report the five-seed mean with a two-sided 95% Student-*t* confidence interval. Random and deterministic controllers were evaluated in five independent blocks of 64 episodes so that uncertainty was summarized on the same block level.

5.2. Baselines and Metrics

The comparison spans two value-factorization baselines, two policy-gradient baselines, random control, and one strong non-learning controller. The VDN sums individual values [27]; QMIX uses a state-conditioned monotonic mixer [9]. IPPO uses local actors and critics, whereas MAPPO uses a centralized critic during training and local actors at execution [28]. POGS-VDN, POGS-IPPO, and POGS-MAPPO retain the proposed coordinator, reward, observation interface, and action mask while replacing the low-level backbone. Vanilla QMIX retains the environment and mask but removes target assignment/features, target-progress reward, convex outage cost, and class balancing.

POGS-Direct-Greedy uses exactly the proposed priority-outage score, greedy one-to-one assignment, charging transition, and return mask but has no learned network: it chooses the feasible movement action whose candidate next position minimizes distance to the assigned target, returns when only return is feasible, and otherwise hovers. Its comparison with POGS-QMIX isolates the contribution of learned low-level control. Greedy versus Hungarian assignment is examined separately with the learned controller in Table 6.

Table 6. POGS-QMIX ablation results (mean $\pm$ sample standard deviation across five independent seeds).

Method	S_H	S_M	S_L	J_H	$O_H^{\max}$	PWEE	F_{ret}
POGS-QMIX	0.547 ± 0.009	0.289 ± 0.010	0.290 ± 0.009	0.877 ± 0.012	38.0 ± 4.7	11.45 ± 0.31	0.00 ± 0.00
Without assignment	0.216 ± 0.011	0.215 ± 0.008	0.213 ± 0.011	0.441 ± 0.037	460.4 ± 20.8	8.30 ± 0.26	0.00 ± 0.00
Without target-progress reward	0.310 ± 0.006	0.262 ± 0.013	0.271 ± 0.011	0.664 ± 0.031	318.6 ± 34.9	8.49 ± 0.29	0.00 ± 0.00
Linear outage ($q = 1$)	0.507 ± 0.019	0.292 ± 0.007	0.290 ± 0.004	0.879 ± 0.013	22.9 ± 3.8	11.02 ± 0.27	0.00 ± 0.00
Hungarian assignment	0.555 ± 0.007	0.298 ± 0.005	0.296 ± 0.005	0.881 ± 0.011	34.7 ± 4.2	11.52 ± 0.29	0.00 ± 0.00
Without action mask	0.267 ± 0.010	0.144 ± 0.009	0.144 ± 0.008	0.834 ± 0.022	321.1 ± 39.5	12.58 ± 0.44	3.76 ± 0.51

The primary metrics are dimensionless coverage fractions S_H , S_M , and S_L ; dimensionless Jain indices J_H , J_M , and J_L ; high-priority maximum outage $O_H^{\max}$ in slots; PWEE in weighted covered-user slots/EU; and failed-return event count F_{ret} per episode. PWEE uses identical fixed priority weights for every method. POGS-QMIX, POGS-MAPPO, POGS-IPPO, and POGS-VDN use the same environment reward, so their raw episode returns are directly comparable. Vanilla QMIX uses an ablated reward formulation and its return is therefore reported separately as a within-method convergence diagnostic.

5.3. Main Comparison and Learning Dynamics

Table 7 shows the multi-seed comparison. POGS-QMIX reaches $S_H = 0.547 \pm 0.009$, exceeding the non-learning POGS-Direct-Greedy controller by 12.6 percentage points. This comparison indicates complementary contributions from structured target assignment and learned low-level control. POGS-QMIX also has the highest mean S_M and S_L , although these absolute margins are small.

The continuity difference is larger: POGS-QMIX reduces $O_H^{\max}$ to 38.0 ± 4.7 slots, versus 102.4 ± 12.6 slots for POGS-Direct-Greedy, a reduction of 62.9%.

The urgency score combines consecutive-outage duration with class-balanced priority, greedy one-to-one matching distributes targets across the fleet, and target-progress feedback supplies a dense motion signal. The reported variants characterize this integrated coordination pipeline and its assignment, progress-shaping, outage-exponent, solver, and

action-mask choices. A matched factorial analysis that varies priority weighting and outage memory separately is reserved for future work.

Table 7. Default-scenario performance (mean $\pm$ standard deviation across five independent seeds/blocks).

Method	S_H	S_M	S_L	J_H	J_M	J_L	$O_H^{\max}$	PWEE	F_{ret}
Vanilla QMIX	0.266 ± 0.014	0.272 ± 0.009	0.270 ± 0.010	0.543 ± 0.031	0.534 ± 0.026	0.534 ± 0.024	466.0 ± 18.2	7.99 ± 0.22	0.00 ± 0.00
POGS-VDN	0.332 ± 0.021	0.276 ± 0.011	0.276 ± 0.012	0.634 ± 0.028	0.531 ± 0.031	0.533 ± 0.029	347.7 ± 31.6	8.62 ± 0.28	0.00 ± 0.00
POGS-IPPO	0.292 ± 0.025	0.275 ± 0.013	0.267 ± 0.014	0.665 ± 0.041	0.630 ± 0.038	0.616 ± 0.040	324.2 ± 37.1	8.01 ± 0.31	0.00 ± 0.00
POGS-MAPPO	0.286 ± 0.022	0.278 ± 0.012	0.277 ± 0.012	0.740 ± 0.035	0.711 ± 0.030	0.707 ± 0.032	252.9 ± 29.4	8.31 ± 0.27	0.00 ± 0.00
POGS-Direct-Greedy	0.421 ± 0.016	0.285 ± 0.008	0.284 ± 0.009	0.815 ± 0.019	0.617 ± 0.022	0.612 ± 0.024	102.4 ± 12.6	9.87 ± 0.19	0.00 ± 0.00
Random	0.256 ± 0.008	0.253 ± 0.007	0.258 ± 0.008	0.644 ± 0.020	0.629 ± 0.018	0.632 ± 0.019	481.6 ± 7.9	7.92 ± 0.12	0.00 ± 0.00
POGS-QMIX	0.547 ± 0.009	0.289 ± 0.010	0.290 ± 0.009	0.877 ± 0.012	0.543 ± 0.025	0.542 ± 0.026	38.0 ± 4.7	11.45 ± 0.31	0.00 ± 0.00

Figure 3 presents the five episode-return trajectories on separate axes and compares the three common evaluation metrics across methods. POGS-QMIX improves after approximately 40,000 steps and stabilizes near 110,000 steps.

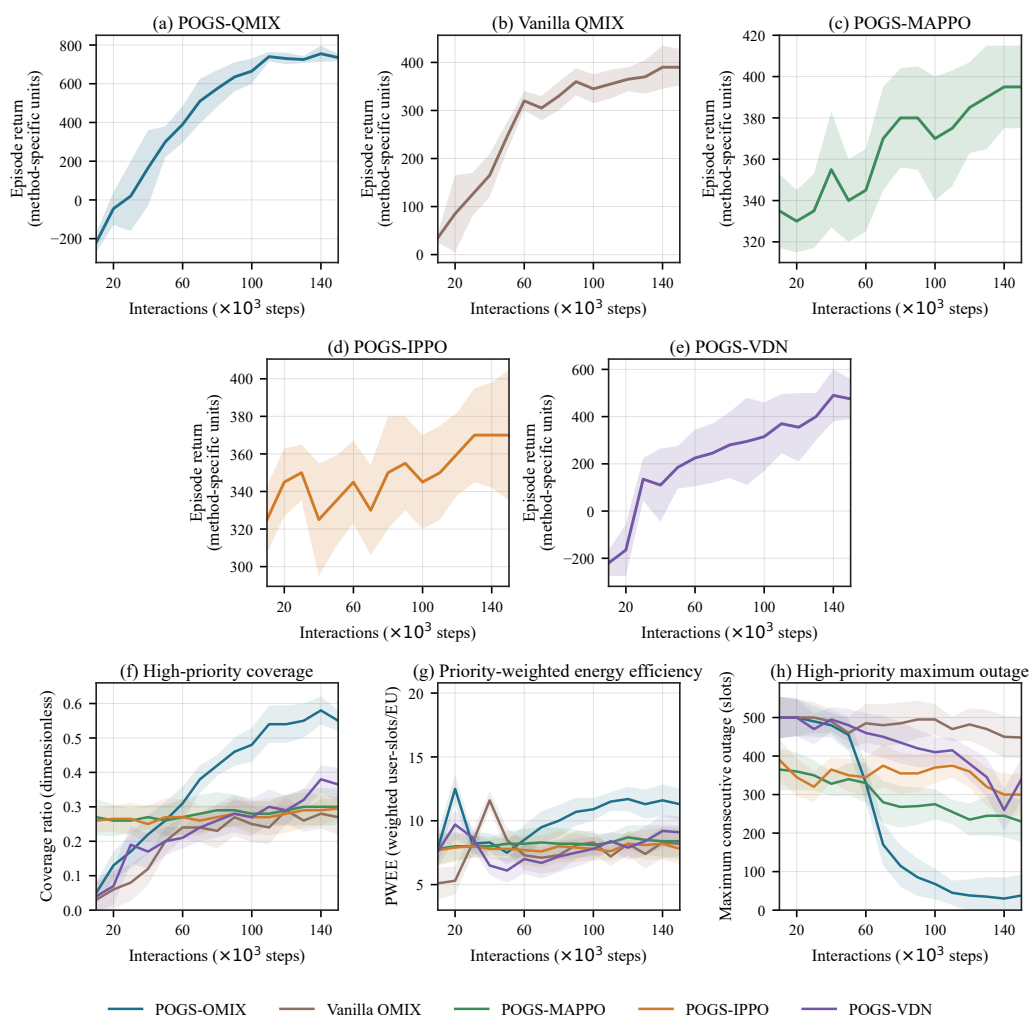


Figure 3. Learning dynamics across five independent training seeds. Panels (a–e) display each method’s episode-return trajectory on a separate axis. The four POGS variants share the same environment reward; Vanilla QMIX uses an ablated reward and is shown as a within-method convergence diagnostic. Panels (f–h) report the common evaluation metrics shared by all methods: high-priority coverage ratio (dimensionless), PWEE (weighted covered-user slots/EU), and high-priority maximum consecutive outage (slots). Lines show five-seed means and shaded bands show two-sided 95% Student-*t* confidence intervals across seed-level means.

5.4. Backbone Interpretation

POGS-QMIX raises J_H from 0.740 to 0.877 and reduces $O_H^{\max}$ from 252.9 to 38.0 slots relative to POGS-MAPPO. Target-conditioned recurrent utilities and replay of rare long-outage transitions are plausible contributors. POGS-MAPPO, in turn, obtains higher medium- and low-priority Jain indices (approximately 0.71 versus 0.54). These results identify a mission-dependent trade-off between high-priority continuity and lower-priority intra-class fairness.

5.5. Ablation Analysis

Five variants were evaluated in the same default scenario and with the same five-seed aggregation as Table 7. The without assignment variant removes target matching and the associated progress term. The without target-progress reward variant retains target features but removes (44). The linear outage variant replaces the quadratic exponent with one. The Hungarian assignment variant replaces greedy pair acceptance with maximum-weight matching while retaining the learned controller. The without action mask variant makes every action available during exploration and value maximization. Figure 4 presents the corresponding class-specific coverage results.

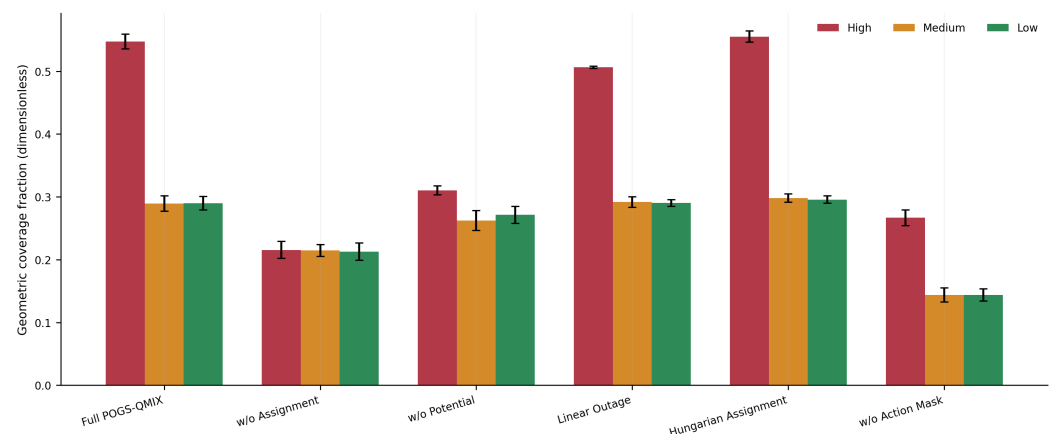


Figure 4. Class-specific geometric coverage fractions in the ablation study (dimensionless; higher is better). Bars are means over five independent seeds and error bars denote two-sided 95% Student- t confidence intervals.

The without-assignment variant produces the largest degradation, reflecting the joint removal of target matching and the associated progress term. Accordingly, this row characterizes the integrated assignment interface, while the without-target-progress row isolates progress shaping within the target-conditioned design. The direct-controller row compares deterministic coordinated actions with learned low-level control. Hungarian matching yields slightly higher mean metrics, while the five-seed 95% confidence intervals overlap. Greedy is retained as the default implementation because it uses a simple sorting-based assignment procedure; Hungarian remains a competitive alternative at the tested fleet scale.

The linear-outage variant obtains the shortest high-priority outage and slightly higher J_H , whereas $q = 2$ obtains higher S_H and PWEE. We retain $q = 2$ as a balance: relative to $q = 1$, it improves delivered high-priority service and weighted service per EU, while $q = 2.5$ provides only a small additional S_H increase, no PWEE gain, and a longer maximum interruption. Applications that prioritize the maximum interruption can instead adopt the linear setting.

Without the action mask, the failed-return count increases from zero to 3.76 ± 0.51 events per episode, while medium- and low-priority coverage are approximately halved. Premature battery depletion reduces recorded energy because fewer UAVs remain operational, which produces an apparent increase in PWEE. PWEE must therefore be interpreted together with failures and outage.

Failure Mechanism Without the Return-Energy Mask

Removing the mask affects more than the return-failure count. During exploration, a coverage-seeking UAV can repeatedly select actions that provide immediate service gains but move it beyond the estimated return-feasible boundary. This can produce a cascade from short-term service gains to battery depletion, reduced fleet capacity, and degraded long-term continuity.

The observed degradation without the mask is consistent with this mechanism.

5.6. Sensitivity and Design Trade-Offs

The sensitivity results characterize the trade-offs among the tested settings. Larger outage weight shortens interruption tails but slightly reduces PWEE; stronger distance and energy-risk penalties reduce delivered service; and the target-progress coefficient has a marked influence. With all other parameters fixed at their default values, 25 EU is the smallest tested reserve yielding zero observed failed returns in the reserve-only sweep of Table 3.

5.7. Operating-Condition Evaluation

We varied four operating dimensions: fleet size $N \in \{4, 5, 6, 7\}$, user population $M \in \{40, 60, 80, 100, 120\}$, coverage radius $R_c \in \{120, 150, 180\}$ m, and UAV speed $v_u \in \{30, 40, 50\}$ m/slot. Only one dimension changed at a time. At every operating point, each learning method was trained from scratch for 150,000 environment steps with five independent seeds. Figures 5–7 therefore report configuration-specific performance after retraining. Deterministic and random controllers used five blocks of 64 evaluation episodes.

The curves in Figure 5 show the highest mean high-priority coverage for POGS-QMIX at all 15 operating points. The medium- and low-priority service amounts remain close to the strongest comparators, with the corresponding intra-class fairness reported separately in Figure 6.

Increasing N from four to seven changes POGS-QMIX high-priority coverage from 0.522 to 0.594. Increasing M from 40 to 120 changes S_H from 0.569 to 0.513. Expanding R_c from 120 to 180 m changes S_H from 0.532 to 0.559, while increasing v_u from 30 to 50 m/slot changes it from 0.529 to 0.551.

Figure 6 complements the coverage results. POGS-QMIX has the highest mean high-priority Jain index at all tested points, but it does not dominate medium- and low-priority fairness. This confirms that fairness improvement is class-specific rather than a general property of the method.

As shown in Figure 7, POGS-QMIX has the lowest mean $O_H^{\max}$ at each operating point, retaining its continuity advantage under both resource-constrained and favorable conditions.

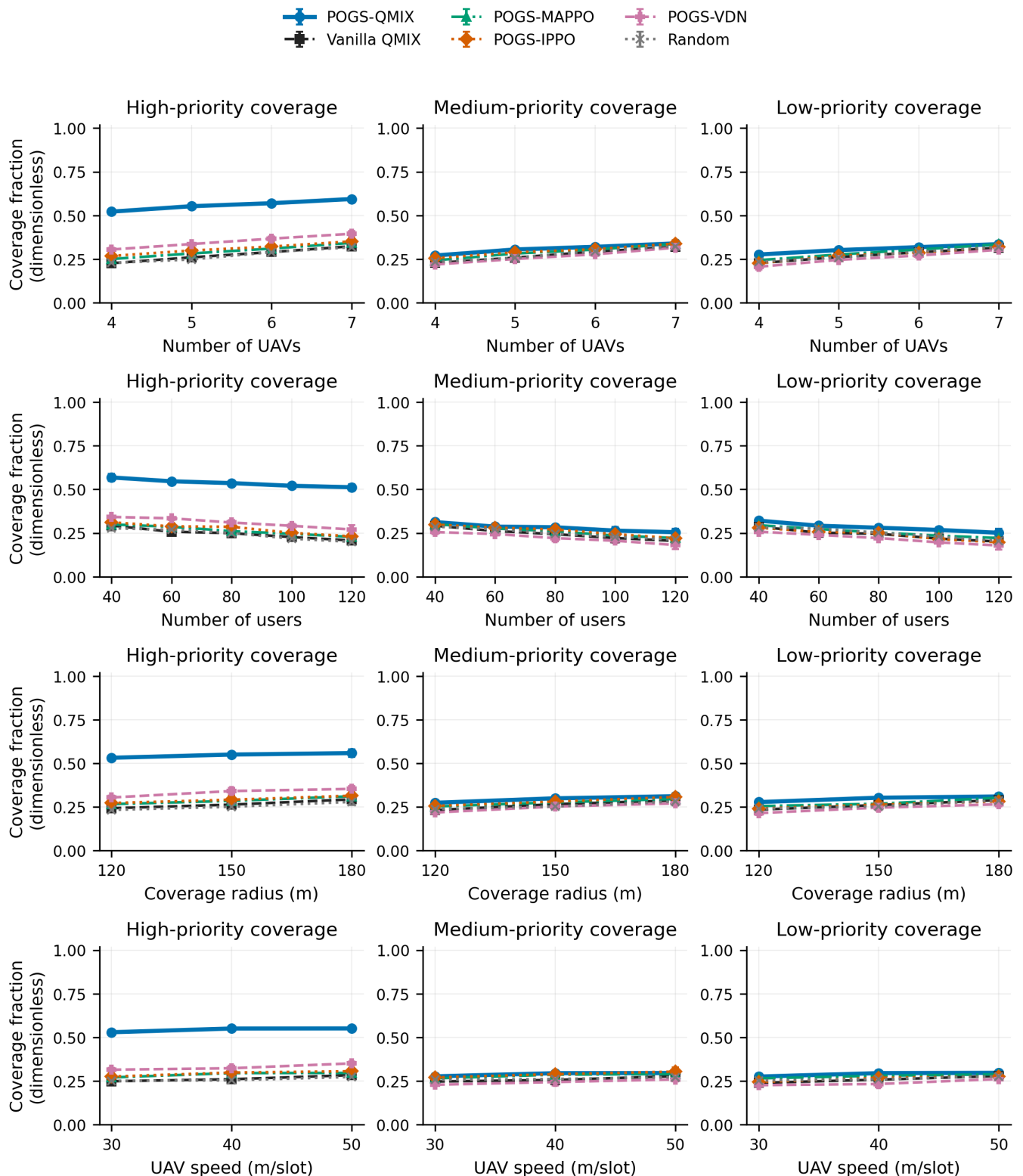


Figure 5. Class-specific geometric coverage fractions (dimensionless) across operating conditions. Rows vary UAV count, user count, coverage radius (m), and UAV displacement (m/slot); columns report high-, medium-, and low-priority coverage. Markers are five-seed means, and error bars denote two-sided 95% Student-*t* confidence intervals.

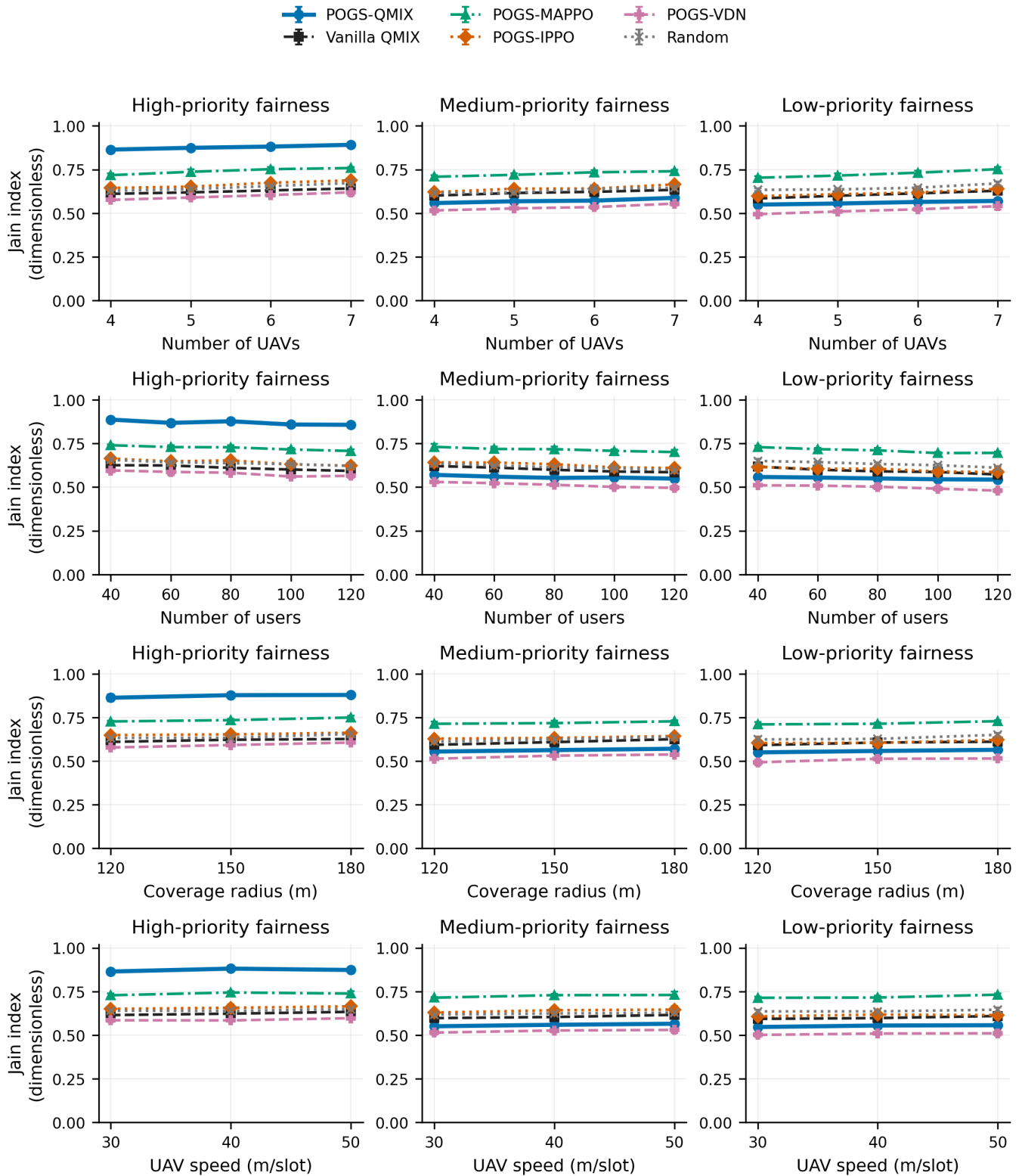


Figure 6. Class-wise Jain fairness (dimensionless) for the four operating-condition sweeps. Fairness was computed separately within each priority class; the corresponding coverage fractions are reported in Figure 5. Markers are five-seed means, and error bars denote two-sided 95% Student-*t* confidence intervals.

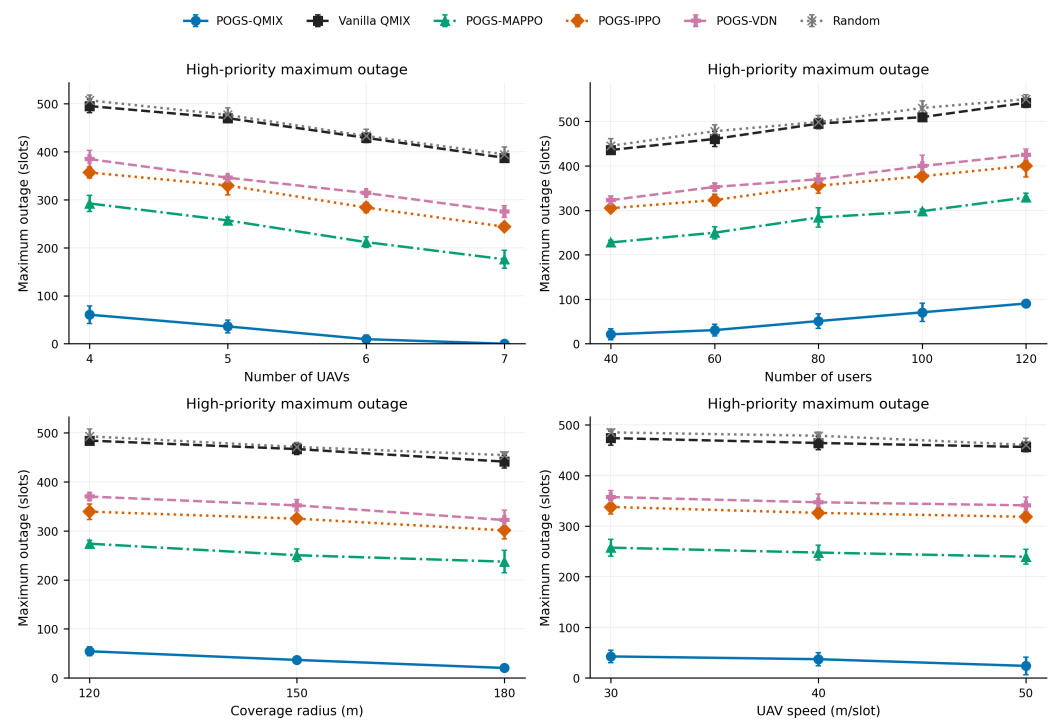


Figure 7. High-priority maximum consecutive outage (slots) across operating conditions. Lower values indicate stronger temporal service continuity; markers are five-seed means, and error bars denote two-sided 95% Student-*t* confidence intervals.

5.8. Energy-Stress Evaluation

The energy-stress study evaluates each of five trained checkpoints over 64 episodes under two conditions: battery capacity reduced from 240 to 180 EU, and charging rate reduced from 15 to 8 EU/slot. Each condition was tested with and without the action mask.

Table 8 shows zero observed failed returns with the mask and more than four events per episode without it, demonstrating the empirical effect of return-energy filtering in the evaluated energy model.

Table 8. Energy-stress evaluation (mean $\pm$ standard deviation across five training seeds; PWEE in weighted covered-user slots/EU and failures in events/episode).

Condition	Mask	S_H	S_M	S_L	PWEE	F_{ret}
$B_{max} = 180$ EU	on	0.528 ± 0.020	0.291 ± 0.011	0.294 ± 0.011	11.32 ± 0.32	0.00 ± 0.00
$B_{max} = 180$ EU	off	0.219 ± 0.028	0.106 ± 0.014	0.106 ± 0.013	13.10 ± 0.49	4.20 ± 0.63
$r_b = 8$ EU/slot	on	0.491 ± 0.022	0.283 ± 0.012	0.280 ± 0.012	10.93 ± 0.35	0.00 ± 0.00
$r_b = 8$ EU/slot	off	0.296 ± 0.026	0.143 ± 0.015	0.140 ± 0.015	13.26 ± 0.47	4.22 ± 0.58

5.9. Experimental Summary

The results quantify multi-seed variability and characterize the integrated target-assignment interface, learned low-level control, outage exponent, assignment solver, and return-energy mask. Linear outage weighting yields the shortest worst interruption, whereas quadratic weighting favors service amount and PWEE. The class-wise results further characterize the allocation trade-off across priority levels.

6. Conclusions

This paper introduced POGS-QMIX for heterogeneous-priority persistent geometric multi-UAV coverage under normalized battery and recharging constraints. The frame-

work couples priority–outage service memory, conflict-reduced target coordination, target-conditioned low-level control, and model-based return-energy filtering around a QMIX backbone. Across five independent seeds, POGS-QMIX achieves high-priority coverage 0.547 ± 0.009 and maximum high-priority outage 38.0 ± 4.7 slots, with zero observed failed-return events under the modeled conditions. The direct-controller and assignment variants characterize performance differences among the integrated coordination and learned-control configurations.

Future work will investigate formal return-risk bounds, adaptive priorities, graph-based scalable representations, and laboratory and outdoor flight validation.

Author Contributions: Conceptualization, H.M. and C.X.; methodology, H.M., C.X. and R.L.; software, H.M.; validation, H.M. and C.X.; formal analysis, H.M. and C.X.; investigation, H.M., C.X. and R.L.; resources, H.M., C.X. and X.L.; writing—original draft preparation, H.M., C.X. and R.L.; writing—review and editing, H.M., C.X., R.L. and X.L.; supervision, C.X. and X.L.; funding acquisition, C.X. 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 62572479.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the 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.

References

1. Merwaday, A.; Güvenç, I. UAV assisted heterogeneous networks for public safety communications. In Proceedings of the IEEE Wireless Communications and Networking Conference Workshops (WCNCW), New Orleans, LA, USA, 9–12 March 2015; pp. 329–334. <https://doi.org/10.1109/WCNCW.2015.7122576>.
2. Mozaffari, M.; Saad, W.; Bennis, M.; Nam, Y.-H.; Debbah, M. A tutorial on UAVs for wireless networks: Applications, challenges, and open problems. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2334–2360. <https://doi.org/10.1109/COMST.2019.2902862>.
3. Hayat, S.; Yanmaz, E.; Muzaffar, R. Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2624–2661. <https://doi.org/10.1109/COMST.2016.2560343>.
4. Cheng, B.; He, M.; Zhu, Z.; He, B.; Chen, J. Development and application of coverage control algorithms: A concise review. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 14906–14927. <https://doi.org/10.1109/TASE.2025.3563489>.
5. Hu, W.; Yu, Y.; Liu, S.; She, C.; Guo, L.; Vucetic, B.; Li, Y. Multi-UAV coverage path planning: A distributed online cooperation method. *IEEE Trans. Veh. Technol.* **2023**, *72*, 11727–11740. <https://doi.org/10.1109/TVT.2023.3266817>.
6. Xiao, J.; Yuan, G.; Xue, Y.; He, J.; Wang, Y.; Zou, Y.; Wang, Z. A deep reinforcement learning based distributed multi-UAV dynamic area coverage algorithm for complex environment. *Neurocomputing* **2024**, *595*, 127904. <https://doi.org/10.1016/j.neucom.2024.127904>.
7. Nigam, N. The multiple unmanned air vehicle persistent surveillance problem: A review. *Machines* **2014**, *2*, 13–72. <https://doi.org/10.3390/machines2010013>.
8. Shakhatareh, H.; Khreishah, A.; Chakareski, J.; Salameh, H.B.; Khalil, I. On the continuous coverage problem for a swarm of UAVs. In Proceedings of the IEEE 37th Sarnoff Symposium, Newark, NJ, USA, 19–21 September 2016; pp. 130–135. <https://doi.org/10.1109/SARNOF.2016.7846742>.
9. Rashid, T.; Samvelyan, M.; de Witt, C.S.; Farquhar, G.; Foerster, J.; Whiteson, S. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; Volume 80, pp. 4295–4304.
10. Scherer, J.; Rinner, B. Persistent multi-UAV surveillance with data latency constraints. *arXiv* **2019**, arXiv:1907.01205.
11. Seyedi, S.; Yazicioglu, Y.; Aksaray, D. Persistent surveillance with energy-constrained UAVs and mobile charging stations. *IFAC-PapersOnLine* **2019**, *52*, 193–198. <https://doi.org/10.1016/j.ifacol.2019.12.157>.
12. Sun, Z.; Wang, N.; Lin, H.; Zhou, X. Persistent coverage of UAVs based on deep reinforcement learning with wonderful life utility. *Neurocomputing* **2023**, *521*, 137–145. <https://doi.org/10.1016/j.neucom.2022.11.091>.

13. Liu, C.H.; Ma, X.; Gao, X.; Tang, J. Distributed energy-efficient multi-UAV navigation for long-term communication coverage by deep reinforcement learning. *IEEE Trans. Mob. Comput.* **2020**, *19*, 1274–1285. <https://doi.org/10.1109/TMC.2019.2908171>.
14. Li, L.; Chen, H. Age of information optimal UAV swarm-assisted sweep coverage in wireless sensor networks. *Ad Hoc Netw.* **2025**, *166*, 103675. <https://doi.org/10.1016/j.adhoc.2024.103675>.
15. Wang, K.; Zhang, X.; Duan, L.; Tie, J. Multi-UAV cooperative trajectory for servicing dynamic demands and charging battery. *IEEE Trans. Mob. Comput.* **2023**, *22*, 1599–1614. <https://doi.org/10.1109/TMC.2021.3110299>.
16. Cheng, X.; Jiang, R.; Sang, H.; Li, G.; He, B. Joint optimization of multi-UAV deployment and user association via deep reinforcement learning for long-term communication coverage. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 5503613. <https://doi.org/10.1109/TIM.2024.3421433>.
17. Luo, X.; Xie, J.; Xiong, L.; Wang, Z.; Liu, Y. UAV-assisted fair communications for multi-pair users: A multi-agent deep reinforcement learning method. *Comput. Netw.* **2024**, *242*, 110277. <https://doi.org/10.1016/j.comnet.2024.110277>.
18. Mathew, N.; Smith, S.L.; Waslander, S.L. Multirobot rendezvous planning for recharging in persistent tasks. *IEEE Trans. Robot.* **2015**, *31*, 128–142. <https://doi.org/10.1109/TRO.2014.2380593>.
19. Arribas, E.; Cholvi, V.; Mancuso, V. Optimizing UAV recharge scheduling for heterogeneous and persistent aerial service. *arXiv* **2022**, arXiv:2205.12656.
20. Datsko, D.; Nekovar, F.; Penicka, R.; Saska, M. Energy-aware multi-UAV coverage mission planning with optimal speed of flight. *IEEE Robot. Autom. Lett.* **2024**, *9*, 2893–2900. <https://doi.org/10.1109/LRA.2024.3358581>.
21. Seong, M.; Jo, O.; Shin, K. Multi-UAV trajectory optimizer: A sustainable system for wireless data harvesting with deep reinforcement learning. *Eng. Appl. Artif. Intell.* **2023**, *120*, 105891. <https://doi.org/10.1016/j.engappai.2023.105891>.
22. Ma, H.; Liu, W.; Yang, G.; Sun, H.; Qu, D.; Yang, H. Efficient charging coverage strategy of UAV based on improved deep reinforcement learning. *IEEE Trans. Veh. Technol.* **2025**, *74*, 11832–11844. <https://doi.org/10.1109/TVT.2025.3556754>.
23. Bai, Y.; Xie, B.; Liu, Y.; Chang, Z.; Jäntti, R. Dynamic UAV deployment in multi-UAV wireless networks: A multimodal-feature-based deep reinforcement learning approach. *IEEE Internet Things J.* **2025**, *12*, 18765–18778. <https://doi.org/10.1109/JIOT.2025.3556300>.
24. Cheng, X.; Jiang, R.; Sang, H.; Li, G.; He, B. Trace pheromone-based energy-efficient UAV dynamic coverage using deep reinforcement learning. *IEEE Trans. Cogn. Commun. Netw.* **2024**, *10*, 1063–1074. <https://doi.org/10.1109/TCCN.2024.3350590>.
25. Chen, D.; Qi, Q.; Fu, Q.; Wang, J.; Liao, J.; Han, Z. Transformer-based reinforcement learning for scalable multi-UAV area coverage. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 10062–10077. <https://doi.org/10.1109/TITS.2024.3358010>.
26. Shi, R.; Yu, X.; Wang, Y.; Tian, Y.; Liu, Z.; Wu, W.; Zhang, X.-P.; Veloso, M.M. Symmetry-informed MARL: A decentralized and cooperative UAV swarm control approach for communication coverage. *IEEE Trans. Mob. Comput.* **2025**, *24*, 8039–8056. <https://doi.org/10.1109/TMC.2025.3553285>.
27. Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W.M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J.Z.; Tuyls, K.; Graepel, T. Value-decomposition networks for cooperative multi-agent learning based on team reward. In Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, Stockholm, Sweden, 10–15 July 2018; pp. 2085–2087.
28. Yu, C.; Velu, A.; Vinitisky, E.; Gao, J.; Wang, Y.; Bayen, A.; Wu, Y. The surprising effectiveness of PPO in cooperative multi-agent games. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24611–24624.

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.