www.nature.com/scientificreports

scientific reports

Check for updates

OPEN

Learning enhanced scheduling and resource allocation for heterogeneous UAV swarms in edge assisted remote sensing

Jingjing Zhang[1], Yunyi Hu[2]✉, Mengmeng Shao[3], You Tang[4], Leilei Wang[5] & Xinyu Li[1]✉

Large-scale 3D mapping and high-resolution remote sensing are essential for environmental monitoring, disaster assessment, and urban planning. Heterogeneous unmanned aerial vehicle (UAV) swarms, equipped with complementary sensing and onboard edge computing capabilities, offer efficient, adaptive, and resource-aware operations. However, achieving complete spatial coverage, ensuring sensing relevance, and optimizing both communication and computational resources remain challenging under dynamic and complex conditions. This paper proposes an energy- and resource-aware cooperative framework, DMMP-PR-TSA, which integrates remote sensing data-driven region partitioning, improved self-organizing map (SOM)-based intelligent pre-assignment, priority-aware dynamic task reallocation (PR), and reinforcement learning (RL)-based task sequence adjustment (TSA). The framework jointly optimizes spatial path planning for sensing tasks and computational resource allocation for edge processing and collaborative task execution, while embedding priority handling to meet deadlines for critical missions. Compared with baseline algorithms, DMMP-PR-TSA demonstrates $15\%-20\%$ higher completion rates in large-scale missions, $10\%-30\%$ improvement under dynamic fleet changes, and consistently higher success rates for high-priority tasks. Simulation results validate its scalability, robustness, and mission-critical applicability, highlighting its effectiveness in advancing the intelligence and operational efficiency of UAV-based large-scale remote sensing and edge-computing-assisted systems.

Keywords Remote sensing data acquisition, Multi-UAV systems, Edge computing, Dynamic scheduling, Resource allocation

In recent years, the introduction of unmanned aerial vehicle (UAV) technology has substantially enhanced the flexibility and efficiency of surveying operations [1,2]. As an emerging platform for mobile mapping and high-resolution remote sensing, UAVs not only improve the spatial accuracy and timeliness of data acquisition but also reduce surveying costs while increasing data accessibility and operational safety [3]. Moreover, with the continuous advancement of onboard computing capabilities, the functional boundaries of edge-intelligent UAVs are being further extended. Modern UAVs are now equipped not only with flexible and efficient data acquisition capabilities, but also with onboard edge computing units that enable in-situ data processing, such as real-time image mosaicking, dense point cloud generation, semantic segmentation, and object detection [4–6]. This paradigm of edge computing not only effectively alleviates the communication and data transmission burdens on the back end but also establishes a solid foundation for adaptive perception, intelligent task offloading, and rapid decision-making throughout the mission process. By performing preliminary filtering, compression, and lightweight analytics of sensed data locally, UAVs can provide low-latency feedback and support distributed cooperative processing for large-scale remote sensing applications, thereby enabling key scenarios such as change detection in disaster response, adaptive data collection in urban monitoring, and real-time collaborative mapping.

[1]School of Computer Science and Engineering, Central South University, Changsha, China. [2]School of Information Resource Management, Renmin University of China, Beijing, China. [3]School of Electronic Information, Central South University, Changsha, China. [4]School of Civil Engineering, Hunan University, Changsha, China. [5]School of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China. ✉email: huyunyi@ruc.edu.cn; lixinyu0826@csu.edu.cn

| https://doi.org/10.1038/s41598-025-34497-z

Despite considerable research progress, most mainstream approaches remain focused on data acquisition or coverage optimization for single UAVs or homogeneous UAV swarms, or treat data collection and edge computing as decoupled problems modeled separately. However, in practical complex applications, large-scale 3D mapping and remote sensing tasks often require heterogeneous UAV swarms to collaboratively execute diverse sensing and edge computing missions. For example, in urban digital twin construction, the system must allocate tasks such as multi-view image capture, LiDAR point cloud scanning, and onboard semantic segmentation according to demand, where these tasks exhibit substantial differences in spatial distribution, data volume, computational complexity, and latency constraints [7,8]. Meanwhile, significant heterogeneity in sensor configurations, onboard computational modules, and energy reserves among UAVs leads to pronounced disparities not only in their data acquisition capabilities but also in their edge computing efficiency. Real-world deployments are further complicated by dynamic task demands and environmental changes, such as real-time shifts in areas of interest, sudden communication bottlenecks, or rapid scene evolution. These factors collectively introduce strong coupling between sensing coverage, onboard processing, and energy–mobility dynamics, making static or uniform task allocation, path planning, or computation scheduling inherently insufficient for achieving real-time, system-level coordination in complex environments.

Existing studies have explored multi-agent path planning, adaptive task allocation, and energy-aware scheduling; however, most remain limited to static resource allocation, homogeneous UAV capabilities, or decoupled optimization of data acquisition and edge processing. To date, a unified theoretical framework and system architecture that can jointly optimize heterogeneous sensing resources, onboard edge computing capabilities, dynamic environments, and adaptive task requirements remains lacking. This gap directly restricts the potential of collaborative edge intelligence and limits system scalability, energy efficiency, and responsiveness in mission-critical remote sensing scenarios. Moreover, the inherent dynamics of remote sensing data acquisition and edge processing environments, as well as the variability of UAV computing and energy resources, often necessitate region re-partitioning and task sequence reallocation in practical deployments.

To address these challenges, this paper focuses on the joint scheduling and resource optimization problem for heterogeneous UAV swarms engaged in collaborative remote sensing data acquisition and edge computing. We propose a learning-enhanced, edge-aware scheduling framework that dynamically partitions and assigns a variety of tasks such as image acquisition, LiDAR scanning, multi-view data collection, real-time image mosaicking, point cloud fusion, semantic segmentation, object detection, and preliminary data filtering to heterogeneous UAV platforms. The framework explicitly models the heterogeneity of tasks with respect to data volume, computational complexity, spatial distribution, and latency constraints, while also considering differences among UAVs in terms of energy reserves, sensor configurations, and onboard edge computing capacity. For example, in disaster emergency remote sensing, the proposed system can efficiently organize UAV swarms to simultaneously perform aerial data acquisition and edge-enabled change detection and damage assessment, thereby enabling high-quality data collection, rapid situational awareness, and adaptive mission response.

Building upon this energy- and edge-aware multi-UAV foundation, this work develops a Dynamic Multi-stage Mission Planning (DMMP) mechanism that integrates three tightly coupled components: (i) a priority-aware, capacity-constrained region partitioning module, (ii) a Self-Organizing Map (SOM)-based intelligent pre-assignment and dynamic task reallocation module, and (iii) a reinforcement-learning-driven adaptive task sequence optimization module. Each module is specifically adapted to the challenges of mixed remote-sensing and onboard edge-processing missions: the partitioning stage embeds energy-hovering-computation constraints into the spatial decomposition, the pre-assignment stage incorporates feasibility-aware matching to ensure resource consistency under dynamic mission events, and the RL-based stage optimizes execution order through a resource-aware Markov decision process. Through this multi-stage, constraint-propagating design, the proposed framework achieves globally consistent scheduling, robust task execution, and efficient multi-type collaboration in complex and dynamic environments.

Extensive simulation results demonstrate that the proposed method achieves significant improvements over mainstream approaches in terms of data coverage, energy utilization, and edge-assisted processing efficiency, thereby providing a solid foundation for the development of next-generation intelligent and scalable UAV-based 3D mapping and remote sensing systems. The main contributions of this work are summarized as follows:

- A unified scheduling and resource optimization framework is proposed for heterogeneous UAV swarms, in which the strong coupling and heterogeneity between large-scale remote sensing data acquisition and edge processing tasks are explicitly modeled. Platform-specific constraints on sensing, computing, and energy are incorporated, and a remote sensing data-driven dynamic region partitioning strategy is employed to ensure both coverage relevance and workload balance.
- A learning-enhanced task allocation and dynamic rescheduling mechanism is developed to address adaptive task coordination under time-varying mission demands and UAV state changes. By embedding type-compatibility and feasibility margins into the matching metric and by leveraging resource-aware RL for sequence planning, sensing and processing tasks are efficiently matched to heterogeneous UAVs, achieving joint optimization of data volume, computational complexity, energy consumption, and latency across large-scale missions.
- Through extensive simulations in large-scale and scalable mapping scenarios, the proposed framework is shown to consistently surpass state-of-the-art baseline algorithms in data coverage, energy utilization, and end-to-end processing efficiency, thereby validating its robustness, scalability, and applicability to intelligent UAV-based remote sensing operations.

Related work

Unmanned aerial vehicles (UAVs) have become integral to the advancement of large-scale remote sensing, three-dimensional (3D) mapping, and automated geospatial data acquisition. In recent years, the field has witnessed substantial research progress spanning UAV-enabled multi-modal data collection, multi-agent coordination, and intelligent information extraction [9–11]. The transition from single-UAV operations to multi-UAV and heterogeneous swarm deployments has markedly improved spatial coverage, data quality, and mission efficiency, while the adoption of onboard edge computing has enabled real-time processing and adaptive system response [12–14]. In parallel, edge-centric intelligence frameworks such as privacy-enforcing data stream processing [15], clustered cohesive edge intelligence for IoT systems [16], and energy-efficient user interaction models for smart environments [17] provide complementary insights into distributed computation and resource-aware decision-making relevant to UAV-assisted sensing systems. Correspondingly, a diverse body of literature has investigated optimal path planning, collaborative task allocation, energy-aware resource management, and distributed edge intelligence, which collectively underpin the algorithmic and theoretical basis for modern UAV-assisted remote sensing and large-scale 3D mapping applications.

Unmanned aerial vehicle (UAV) technology has fundamentally transformed large-scale remote sensing and three-dimensional (3D) mapping, enabling high-resolution, flexible, and cost-effective spatial data acquisition [18,19]. Early research primarily focused on data collection and photogrammetric modeling using single UAVs in urban and environmental applications [20]. With increasing application demands, collaborative multi-UAV systems have gradually become a prominent research focus. Recent studies have investigated coordinated area coverage, multi-agent path planning, and spatial data redundancy optimization to enhance mission efficiency and mapping accuracy. Ao et al. [21] introduced multi-agent deep reinforcement learning (MADRL) into collaborative trajectory optimization for multi-UAV systems, achieving joint optimization of energy consumption, coverage efficiency, and network connectivity. Westheider et al. [22] employed deep reinforcement learning for adaptive path planning and regional task assignment, significantly improving cooperation efficiency. Fu et al. [23] proposed a hierarchical planning approach with a mixture-of-experts mechanism to support energy-constrained agricultural remote sensing. Recent studies on spiking neural networks for intelligent edge computing [24] further highlight the growing relevance of low-power, neuromorphic processing for UAV-based remote sensing platforms. Overall, UAV swarms demonstrate strong advantages in large-scale remote sensing and 3D mapping due to rapid deployment, flexible mobility, and multi-view perception.

With the rapid development of edge computing and artificial intelligence, UAVs equipped with onboard computational resources—also referred to as edge-intelligent UAVs—have attracted widespread attention for real-time task processing. Unlike traditional workflows relying on centralized ground-station processing, edge-intelligent UAVs can perform computation-intensive tasks such as image fusion, semantic segmentation, object detection, and change detection in real time during flight [25–27]. This reduces communication burdens, alleviates bandwidth constraints, and enhances system responsiveness [2]. Ongoing research continues to advance UAV edge intelligence, including multi-target detection [28], lightweight fire/smoke recognition [29,30], and precision ecological monitoring [31]. Duan et al. [32] achieved centimeter-level accuracy in road crack detection through real-time semantic segmentation. Meanwhile, multi-platform cooperative frameworks such as UAV–UGV-based urban monitoring [33] and RIS-assisted emergency response [34] demonstrate the importance of resilient coordination and communication in complex environments. Overall, edge-intelligent UAVs substantially enhance the autonomy, responsiveness, and real-time decision-making capacity of remote sensing systems.

Despite these advances, most existing works optimize sensing coverage or computation efficiency in isolation, without jointly considering heterogeneous sensing modalities, onboard computation demands, and dynamic task evolution in large-scale missions [35]. Real-world deployments involve significant heterogeneity across sensing tasks (e.g., imagery, LiDAR, multispectral), edge-processing workloads (e.g., segmentation, detection, fusion), and UAV platform capabilities. For example, Gao et al. [36] designed an AoI-aware UAV sensing method for multi-point data collection. Wan et al. [37] utilized deep reinforcement learning for emergency scheduling in disaster scenarios. Liao et al. [38] and Wang et al. [39] studied multi-platform collaborative computing in heterogeneous UAV–vehicle and UAV–surface-vessel systems. Raivi et al. [40] developed a multi-agent cooperative scheme for post-disaster IoT data aggregation, while Huang et al. [41] applied federated DRL for efficient caching and offloading in UAV-assisted vehicular networks. Zhang et al. [42] and Wang et al. [43] further explored joint trajectory and resource optimization in heterogeneous D2D and UAV–surface hybrid networks. In addition, cooperative UAV routing strategies for urban environments [44] highlight the importance of scalable, communication-aware coordination for large UAV networks.

Although substantial progress has been made, the deep integration of heterogeneous large-scale sensing and intelligent onboard processing remains challenging. Real deployments require systems capable of covering wide geographic areas, handling multi-modal high-resolution data, and adapting to real-time task variations and platform dynamics. UAV swarms differ in sensing capabilities, computing modules, and energy characteristics; meanwhile, tasks vary in priority, data volume, and computational complexity. Operational environments often involve shifting regions of interest, abrupt requirement changes, and fluctuating UAV states, further complicating system coordination. Thus, achieving unified modeling and efficient collaborative optimization across heterogeneous tasks, heterogeneous platforms, and dynamic resource constraints remains a fundamental scientific challenge in large-scale remote sensing and 3D mapping.

Materials and methods

In multi-task UAV-assisted remote sensing scenarios, the operational area often spans a large geographical range, exhibiting heterogeneous surface characteristics and a non-uniform distribution of task priorities. Directly employing a global cruising strategy not only induces redundant coverage and excessive travel distance, but also leads to inefficient utilization of onboard resources, including energy, computational capacity, and communication bandwidth. To address these challenges, we propose a Dynamic Multi-stage Mission Planning (DMMP) UAV task planning framework that seamlessly integrates priority-aware capacity-constrained region partitioning, feasibility-aware dynamic task reallocation, and reinforcement learning-based task sequence optimization, aiming to jointly optimize task completion ratio, execution latency, and energy consumption under dynamic mission environments. In the following, the three stages are referred to as the D (partitioning), PR (preallocation and reallocation), and TSA (task sequence adjustment) modules, forming the overall DMMP-PR-TSA framework.

In the initialization phase, the framework performs remote sensing feature extraction and task priority determination to filter out low-relevance regions. Subsequently, under multi-dimensional UAV capacity constraints, a capacity-constrained power diagram partitioning combined with local refinement is employed to generate spatially compact, workload-balanced, and priority-aware task subregions, thereby maximizing the utilization efficiency of sensing and computation resources. This partitioning stage provides the D module of DMMP, yielding per-UAV coverage regions that serve as the initial task sets for subsequent PR and TSA stages.

During mission execution, when dynamic events such as task addition, location change, priority adjustment, or UAV failure occur, a dynamic task reallocation mechanism is triggered to promptly update the UAV–task assignment, reducing resource conflicts and minimizing redundant travel. Based on the updated assignments, a reinforcement learning-driven dynamic task sequencing strategy further refines the execution order, ensuring that high-priority tasks are completed first while balancing workload among multiple UAVs and mitigating path conflicts. These two stages correspond to the PR and TSA modules in the DMMP framework, and operate on the task sets and residual capacities produced by the partitioning stage.

This multi-stage coordination paradigm enables the proposed framework to adaptively respond to environmental changes and resource fluctuations, ensuring robust and efficient performance in complex UAV remote sensing missions. The subsequent subsections detail each of the DMMP-PR-TSA modules, including the capacity-constrained region partitioning methodology (D), the dynamic task reallocation strategy (PR), and the reinforcement learning-based sequencing mechanism (TSA), and clarify how their inputs and outputs are mathematically linked.

Capacity-constrained task region partitioning in UAV-assisted remote sensing

In UAV-assisted remote sensing missions with onboard computing, where the operational area typically spans a large geographic range with heterogeneous ROIs, directly dispatching each UAV to cover the entire area leads to redundant coverage, excessive travel distance, and inefficient utilization of sensing, computing, and communication resources. To address this, the proposed capacity-constrained task region partitioning mechanism processes remote sensing data to extract task-relevant features and determine ROI priorities, while incorporating heterogeneous UAV capacity profiles—including residual energy, allowable hovering duration, onboard computational capability, and communication bandwidth—into the partitioning process. By jointly considering spatial task distribution and platform-specific constraints, the method allocates each UAV to a spatially compact, workload-balanced, and priority-aware subregion, thereby reducing inter-UAV interference, minimizing mission completion time, and providing a foundation for localized computation–communication scheduling in subsequent stages. In the context of DMMP, this stage defines the initial region-level task sets for each UAV.

Problem notation and capacity constraints

Let the sensing area $\mathcal{A}$ be discretized into a set of grid cells $\mathcal{G} = \{g_1, \ldots, g_N\}$ of resolution Δr. The UAV set is $\mathcal{U} = \{1, \ldots, U\}$, where UAV *u* is located at $\mathrm{p}_u(t)$ at time *t*. Each UAV is characterized by a three-dimensional capability vector:

$$\mathrm{C}_u(t) \triangleq \Big\{C_{u,E}(t),\ C_{u,H}(t),\ C_{u,F}(t)\Big\}, \tag{1}$$

where $C_{u,E}(t)$ denotes the remaining energy budget (J), $C_{u,H}(t)$ the available hovering time (s), and $C_{u,F}(t)$ the available onboard computational capacity (FLOPs).

Similarly, each grid cell g_i is associated with a workload vector:

$$\boldsymbol{\omega}(g_i) \triangleq \Big\{\omega_{E,i},\ \omega_{H,i},\ \omega_{F,i}\Big\}, \tag{2}$$

where $\omega_{E,i}$ is the energy cost to reach and sense g_i, $\omega_{H,i}$ the hovering time required for data capture, and $\omega_{F,i}$ the computation required for preliminary image processing (e.g., calibration, orthorectification, feature extraction). The priority weight w_i^{pri} indicates the sensing importance of g_i, determined during the priority stratification stage according to extracted feature relevance and mission objectives.

Capacity-constrained region partitioning algorithm

The proposed capacity-constrained partitioning framework first processes raw remote sensing data to extract task-relevant features (e.g., label-derived change probability), filtering out low-relevance areas via a predefined threshold. The remaining high-relevance areas are stratified into multiple priority levels. Grid cells are then assigned to UAVs using a capacity-constrained power diagram-based optimization, followed by local refinement to enhance spatial contiguity. The resulting operational map allocates each UAV a compact, priority-aware coverage region that balances workload and supports localized scheduling in subsequent mission stages. The overall workflow of the proposed capacity-constrained region partitioning algorithm is illustrated in Fig. 1.

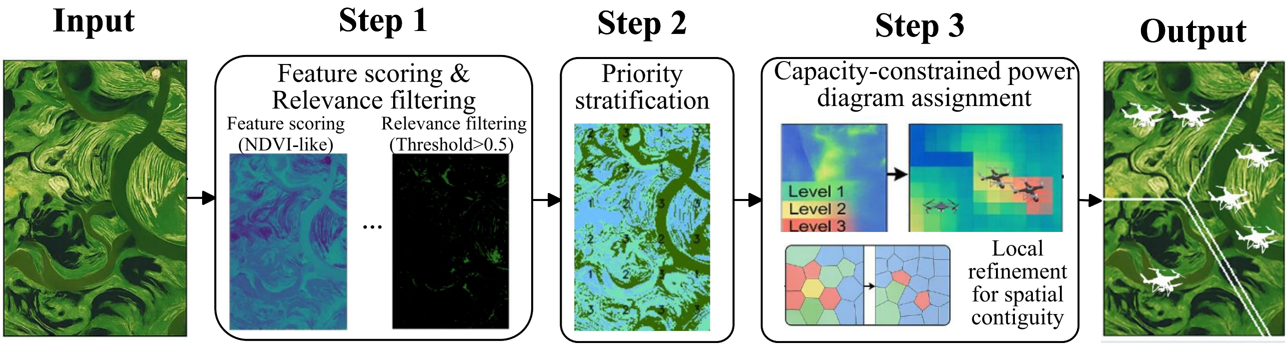


Fig. 1. Framework of the proposed priority-aware capacity-constrained task region partitioning for UAV-assisted remote sensing.

The partitioning problem is formulated as a capacity-constrained spatial clustering problem, where each grid cell is assigned to exactly one UAV. The objective is to minimize the total travel and processing cost while satisfying UAV capacity constraints in (*E*, *H*, *F*) dimensions and promoting contiguous coverage regions. Let $x_{ui} \in \{0, 1\}$ denote the binary assignment variable; the optimization model is:

$$\begin{aligned} \min_{\{x_{ui}\}} & \sum_{i\in\mathcal{G}}\sum_{u\in\mathcal{U}} \left(\alpha\, d(\mathrm{p}_u(t), g_i) - \gamma\, w_i^{\mathrm{pri}}\right) x_{ui} \\ & + \lambda_{\mathrm{TV}} \sum_{(i,j)\in\mathcal{E}}\sum_{u\in\mathcal{U}} |x_{ui} - x_{uj}|, \end{aligned} \tag{3}$$

where α balances travel distance cost, γ emphasizes allocation to high-priority cells, and λ_{TV} controls the total variation regularization to improve spatial compactness.

Subject to:

$$\sum_{u\in\mathcal{U}} x_{ui} = 1, \quad \forall i \in \mathcal{G}, \tag{4}$$

$$\sum_{i\in\mathcal{G}} \omega_{k,i} x_{ui} \le C_{u,k}(t), \quad \forall u \in \mathcal{U},\ \forall k \in \{E, H, F\}, \tag{5}$$

$$x_{ui} \in \{0, 1\}, \tag{6}$$

where $\mathcal{E}$ is the adjacency set of neighboring cells used in the TV term. The resulting assignment variables $\{x_{ui}\}$ induce, for each UAV *u*, a region-level task set $\mathcal{G}_u \triangleq \{g_i \in \mathcal{G} \mid x_{ui} = 1\}$, which serves as the input task set for the subsequent multi-stage planning modules.

The optimization is solved via a capacity-constrained power diagram algorithm. Lagrange multipliers $\mu_{u,k}$ $(\forall u \in \mathcal{U},\ k \in \{E, H, F\})$ are introduced for each UAV–capacity pair. At each iteration, the generalized cost is computed as:

$$\delta_{ui} = \alpha\, d(\mathrm{p}_u(t), g_i) - \gamma\, w_i^{\mathrm{pri}} + \sum_{k\in\{E,H,F\}} \mu_{u,k}\, \omega_{k,i}, \tag{7}$$

and each g_i is assigned to the UAV with the smallest δ_{ui}. The multipliers are updated via:

$$\mu_{u,k} \leftarrow \left[\mu_{u,k} + \rho\left(\sum_{i\in\mathcal{G}} \omega_{k,i} x_{ui} - C_{u,k}(t)\right)\right]_+, \tag{8}$$

where ρ is the subgradient step size, and $[\cdot]_+$ denotes projection onto the non-negative reals. After each assignment, local 1-swap operations reduce the TV term, ensuring spatially compact partitions. From the DMMP perspective, this stage thus produces the initial feasible task sets $\{\mathcal{G}_u\}$ and associated capacity margins, which are then refined and updated by the PR and TSA modules under dynamic mission conditions.

To account for environmental dynamics and UAV motion, partitioning is periodically re-evaluated every Δt seconds. UAV positions are predicted via a constant-velocity Kalman filter (CV-KF), capacities are updated based on executed workloads, and a hysteresis threshold ε is applied to prevent unnecessary reassignments when the objective change ΔJ is small. A maximum-change budget *B* limits the number of cells that may be reassigned in each update. The next subsection builds upon this partitioning outcome and introduces the dynamic task reallocation strategy, enabling adaptive response to time-varying mission demands and platform state variations.

Multi-stage task planning for UAV-based remote sensing and edge computing

In heterogeneous UAV-assisted remote sensing missions with onboard edge computing, sensing tasks are often characterized by diverse spatial priorities, varying data acquisition workloads, heterogeneous onboard processing demands, and strict temporal deadlines. UAV platforms exhibit significant differences in sensing payload configurations, real-time processing capabilities, flight endurance, and communication bandwidth. The dynamic nature of the remote sensing operational theater, including the arrival of new tasks, updates to task locations, changes in sensing priorities, and potential UAV platform failures, further amplifies the challenge of achieving efficient and reliable mission execution. These complexities render static or single-phase allocation strategies inadequate for meeting performance goals such as maximizing remote sensing task completion ratio, minimizing overall mission latency, and reducing end-to-end energy consumption.

To address these challenges, this work develops a capacity-constrained multi-stage task planning framework within the DMMP architecture that integrates initial task preallocation, adaptive task reallocation, and reinforcement learning-driven execution sequence optimization. Building directly on the region-level task sets $\{\mathcal{G}_u\}$ produced by the D module, the PR module constructs UAV-specific task lists $\{\mathcal{T}_u^{\mathrm{PR}}\}$ under feasibility constraints, and the TSA module further optimizes the execution order of tasks within each $\mathcal{T}_u^{\mathrm{PR}}$. In the preallocation stage, remote sensing tasks are initially matched to UAVs by jointly considering heterogeneous platform capabilities and mission-specific sensing–processing requirements, ensuring balanced workloads and modality compatibility. The adaptive reallocation stage reacts to operational dynamics by remapping tasks to available UAVs in real time, mitigating resource contention and avoiding redundant flight paths or duplicated onboard processing. Finally, the execution sequence optimization stage employs reinforcement learning to refine task ordering, ensuring that high-priority sensing tasks are completed first while maintaining balanced UAV utilization and minimizing inter-platform trajectory conflicts.

The proposed approach distinguishes itself through its multi-stage coordination strategy tailored to heterogeneous UAV remote sensing systems, explicit modeling of platform-specific sensing and computing constraints, and robust performance under dynamic and uncertain mission conditions. In particular, the DMMP-PR-TSA framework provides a mathematically linked pipeline, where partitioning, preallocation/reallocation, and sequence optimization share common capacity and feasibility descriptors rather than operating as independent heuristics. The remainder of this section is organized as follows: "Problem analysis" analyzes the operational constraints in UAV remote sensing and edge computing scenarios, Section ?? formulates the task planning problem as a constrained optimization model, and Section ?? and "RL-based task sequence adjustment algorithm for UAV remote sensing" presents the proposed algorithmic solution in detail.

Problem analysis

Constraint analysis in UAV-assisted remote sensing data collection and edge intelligence processing missions

In UAV-assisted remote sensing missions integrated with onboard or cooperative edge intelligence processing, task execution is constrained by a set of physical and operational factors. These include UAV flight endurance, onboard computing capability, task priority ordering, temporal deadlines, UAV-task type compatibility, and precedence requirements. Such constraints fundamentally influence the feasibility, timeliness, and efficiency of remote sensing data collection and subsequent processing. The formal definitions are as follows.

(1) UAV flight endurance constraint: In large-scale remote sensing operations, certain task points may be located at considerable distances from the UAV's current position. Let p_u denote the position of UAV *u* and p_i the position of task *i*. The maximum flight range of UAV *u* is $D_u^{\max}$. The endurance constraint is defined as:

$$\|\mathrm{p}_u - \mathrm{p}_i\| \leq D_u^{\max}, \quad \forall u \in \mathcal{U},\ \forall i \in \mathcal{T}_u. \tag{9}$$

This constraint is not only determined by the UAV's energy capacity but also serves as an intuitive manifestation of energy-aware planning, ensuring that distant remote sensing tasks can be reached and completed without depleting the UAV's battery before mission completion.

(2) UAV computing capability constraint: Prior to assigning remote sensing data processing tasks to a UAV, the feasibility of real-time execution must be evaluated, considering both the onboard computational unit and the availability of cooperative edge servers. Let $F_u^{\max}$ denote the maximum computational capacity of UAV *u*, and $\sum_{i\in\mathcal{T}_u} f_i$ its total assigned processing workload. The constraint is:

$$\sum_{i\in\mathcal{T}_u} f_i \leq F_u^{\max}, \quad \forall u \in \mathcal{U}. \tag{10}$$

This ensures that the UAV processes only workloads that can be completed within operational time windows, thereby preventing mission delays.

(3) Task priority constraint: Let p_i^{pri} represent the priority of task *i*. Higher-priority tasks–such as urgent disaster-monitoring data acquisition or real-time edge analysis–must be scheduled before lower-priority tasks (e.g., periodic environmental mapping or deferred batch processing). The constraint is expressed as:

$$p_i^{\mathrm{pri}} > p_j^{\mathrm{pri}} \;\Rightarrow\; i \prec_u j, \quad \forall u \in \mathcal{U}. \tag{11}$$

This guarantees that mission-critical sensing and processing tasks are executed first, even under limited resource availability.

(4) Task deadline constraint: Let $t_{\text{start},i}$ and $t_{\text{end},i}$ denote the start and completion times of task *i*, and D_i its hard deadline. The constraint is:

$$t_{\text{end},i} - t_{\text{start},i} \leq D_i, \quad \forall i \in \mathcal{T}. \tag{12}$$

This ensures both data acquisition and subsequent processing are completed within their respective time limits, which is crucial for near-real-time applications.

(5) UAV-task type compatibility constraint: Let $\phi_{u,i} \in \{-1, 0, 1\}$ indicate the compatibility of UAV *u* with task *i* (e.g., acquisition-only, processing-capable, or integrated acquisition–processing). UAVs without onboard computing units typically have longer flight endurance and greater hovering capability, making them more suitable for extensive data collection missions. Conversely, UAVs with processing units are prioritized for tasks requiring immediate edge intelligence processing. The constraint is:

$$\phi_{u,i} = 1 \;\Rightarrow\; i \in \mathcal{T}_u. \tag{13}$$

(6) Precedence constraint: For a task sequence π_u assigned to UAV *u*, if task *i* must be executed before task *j*, the precedence relation is:

$$i \prec_u j \;\Leftrightarrow\; \pi_u(i) < \pi_u(j), \quad \forall u \in \mathcal{U}. \tag{14}$$

For instance, in missions where high-resolution imagery must be collected prior to initiating onboard or edge-based object detection, the acquisition task must precede the processing task to maintain workflow integrity.

Scenarios triggering joint reallocation and task-sequence dynamic adjustment in uav remote sensing with edge computing

This section In UAV-assisted remote sensing missions equipped with onboard or cooperative edge computing capabilities, task execution may encounter dynamic changes in mission requirements, spatial distribution of task points, or UAV operational states. Such changes may arise from real-time onboard analysis, refined geospatial information, or unexpected platform failures. These events can trigger two coordinated mechanisms: (i) joint reallocation of tasks among UAVs, and (ii) dynamic adjustment of the task sequence for the affected UAVs. highlights representative illustrative examples where such mechanisms are activated, ensuring timely completion of both data acquisition and processing while satisfying operational constraints defined in "Problem analysis".

(1) New urgent task triggered by real-time remote sensing analysis: This example demonstrates how the system reacts to a newly emerging high-priority task detected through real-time onboard analysis. During large-scale remote sensing operations, UAVs equipped with onboard processors may conduct real-time analysis of captured imagery, such as change detection in disaster zones, rapid vegetation index assessment, or moving object tracking. Suppose the pre-planned task sequence for UAV *u* is:

$$\tau_u \equiv \tau(\text{TASK}_1, \text{TASK}_2, \ldots, \text{TASK}_i, \text{TASK}_j, \ldots, \text{TASK}_m),$$

and UAV *u* is currently performing TASK_i. If onboard or ground-based processing detects a new task TASK_k —for example, identifying a newly flooded area or an unauthorized construction site—with a higher priority than the next scheduled task, i.e.,

$$pri_k > pri_j,$$

then TASK_k is inserted into the sequence immediately after TASK_i:

$$\tau_u \Leftarrow \{\text{TASK}_i \rightarrow \text{TASK}_k \rightarrow \text{TASK}_j\}.$$

This update is executed only if it satisfies the endurance, deadline, and computing capacity constraints in Section 3.3. Figure 2 visually illustrates this insertion process and the resulting change in the UAV's local mission plan.

(2) Task location update triggered by refined geospatial information: This example highlights how task ownership is reassigned when refined geospatial analysis alters the optimal UAV-task mapping. As UAVs collect multi-angle images, subsequent photogrammetric processing, image registration, or target tracking may update the coordinates of pending tasks with improved precision. For example, in forest fire monitoring, smoke plume tracking may cause the location of a high-priority sensing point to shift as the fire front advances. If UAV *u* is performing TASK_i and the updated position of TASK_j significantly increases the remaining travel distance or violates UAV *u*'s residual endurance constraint, TASK_j is reassigned to another

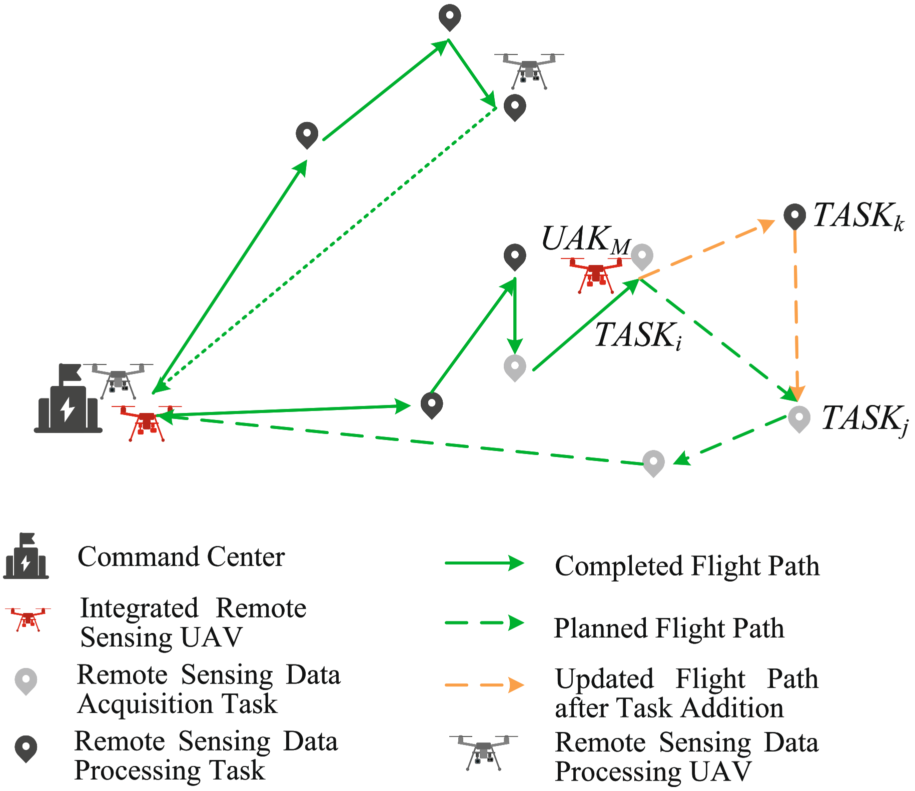

TASK_k
UAK_M
TASK_i
TASK_j
Command Center
Integrated Remote Sensing UAV
Remote Sensing Data Acquisition Task
Remote Sensing Data Processing Task
Completed Flight Path
Planned Flight Path
Updated Flight Path after Task Addition
Remote Sensing Data Processing UAV

Fig. 2. Illustrative example of joint task reallocation and sequence adjustment triggered by a newly detected urgent remote sensing task (e.g., sudden flood monitoring point identified during ongoing aerial survey).

UAV u_2. Figure 3 provides an example of this reassignment triggered by wildfire front relocation, showing how spatial updates propagate through the planning process.

(3) UAV failure during execution with remaining remote sensing tasks: This example illustrates the system's fault-tolerance mechanism when a UAV becomes inoperable mid-mission. In real-world field operations, UAVs may experience unexpected hardware or communication failures. Suppose UAV u becomes inoperable immediately after completing TASK_j. The remaining tasks, which may include critical sensing operations, are redistributed among other UAVs while preserving precedence and deadline feasibility. This example demonstrates how DMMP-PR-TSA maintains mission continuity even under platform loss, which is essential for high-stakes or time-sensitive aerial sensing deployments.

Scenarios triggering only task-sequence dynamic adjustment

In addition to the cases requiring both task reallocation and sequence adjustment, there are scenarios where only the execution order of tasks assigned to a single UAV needs to be updated. Such situations occur when multiple pending tasks of the UAV experience changes in their relative urgency or execution priority without requiring reassignment to other UAVs.

For example, the priority of a low-priority remote sensing task may increase due to newly acquired environmental data or updated mission objectives. In this case, the task-sequence dynamic adjustment mechanism reorders the remaining tasks in the UAV's sequence τ_u based on the updated priorities, time-window constraints, and spatial distribution, thereby enhancing mission completion efficiency while minimizing disruption to ongoing operations. This example demonstrates how the TSA module alone can adapt local execution order without invoking global task reassignment.

Network structure and input representation

To capture both spatial and functional heterogeneity, we employ an enhanced Self-Organizing Map (SOM) neural network consisting of an input layer and a two-dimensional competitive layer, as illustrated in Fig. 4. The input layer contains M nodes, each corresponding to a task $TASK_i$. The competitive layer is structured as a $5 \times U$ neuron grid, where each neuron represents a UAV and its competitive neighborhood.

For each task $TASK_i$, the input feature vector is defined as

$$\mathrm{INFO}_{TASK_i} = (\mathrm{POS}_{TASK_i}, \phi_{TASK_i}, \mathrm{RES}_{TASK_i}), \tag{15}$$

where $\mathrm{POS}_{TASK_i} = (X_i, Y_i)$ denotes the geographical coordinates of the task, ϕ_{TASK_i} indicates the sensing type required (e.g., electro-optical, multispectral, LiDAR), and $\mathrm{RES}_{TASK_i} = (t_i^{\mathrm{req}}, c_i^{\mathrm{req}})$ specifies the required sensing duration t_i^{req} and processing workload c_i^{req} in CPU cycles.

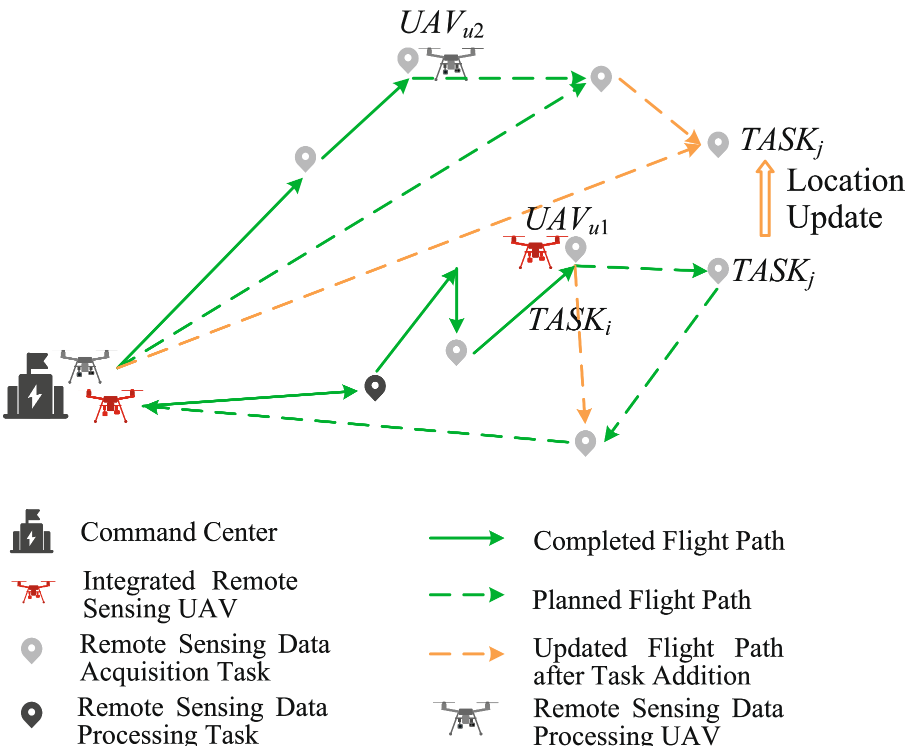

UAV_{u2}
$TASK_j$
Location Update
UAV_{u1}
$TASK_j$
$TASK_i$
Command Center
Integrated Remote Sensing UAV
Remote Sensing Data Acquisition Task
Remote Sensing Data Processing Task
Completed Flight Path
Planned Flight Path
Updated Flight Path after Task Addition
Remote Sensing Data Processing UAV

Fig. 3. Illustrative example of joint task reallocation and sequence adjustment triggered by an updated task location (e.g., wildfire front relocation in ongoing aerial monitoring).

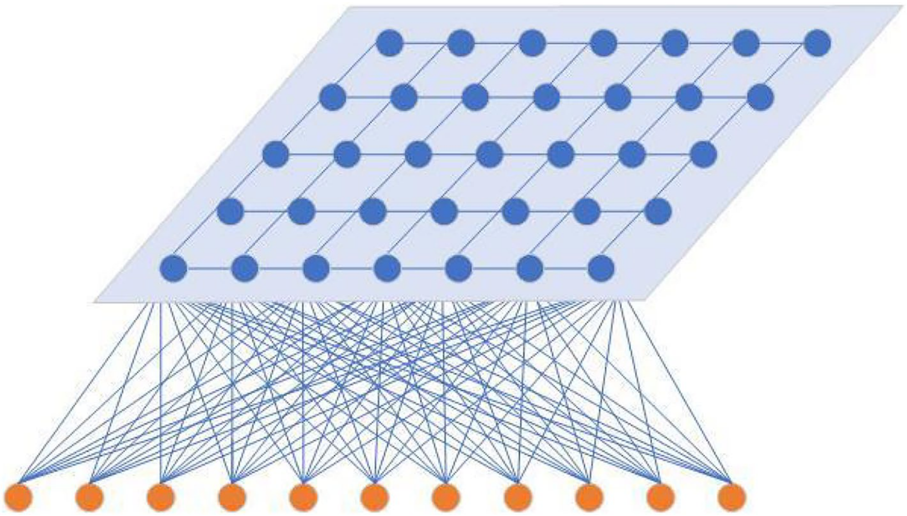


Fig. 4. SOM network structure for UAV remote sensing task allocation.

Similarly, each UAV UAV_u is represented by:

$$\mathrm{INFO}_{UAV_u} = (\mathrm{POS}_{UAV_u}, \psi_{UAV_u}, \mathrm{RES}_{UAV_u}), \tag{16}$$

where $\mathrm{POS}_{UAV_u} = (X_u, Y_u)$ is its current position, ψ_{UAV_u} encodes its sensor/computation capability type, and $\mathrm{RES}_{UAV_u} = (T_{\mathrm{re}}^{(UAV_u)}, C_{\mathrm{re}}^{(UAV_u)})$ denotes the residual flight time and available computational capacity.

Matching distance metric

The SOM competitive layer determines the best-matching UAV for each task by minimizing a *matching distance* that integrates spatial separation, capability compatibility, and resource feasibility. The overall matching distance between $TASK_i$ and UAV_u is:

$$D_{\text{INFO}}^{(i,u)} = D_p + c_\phi D_\phi + c_{\text{RES}} D_{\text{RES}}, \tag{17}$$

where:

(1) The *spatial distance* term,

$$D_p = (X_i - X_u)^2 + (Y_i - Y_u)^2, \tag{18}$$

measures the squared Euclidean distance between task and UAV coordinates, promoting assignments with lower travel cost.

(2) The *capability mismatch penalty*,

$$D_\phi = \begin{cases} 0, & |\psi_{UAV_u} - \phi_{TASK_i}| \leq 1, \\ \infty, & \text{otherwise}, \end{cases} \tag{19}$$

ensures that only UAVs with compatible sensing types are considered; incompatible matches incur an infinite penalty, effectively removing them from the candidate set.

(3) The *resource feasibility term*,

$$D_{\text{RES}} = \Delta_{\text{time}} + \Delta_{\text{comp}}, \tag{20}$$

accounts for both time and computation resource margins. The time margin penalty is:

$$\Delta_{\text{time}} = \begin{cases} \exp\{-c_{\text{time}} \cdot \text{diff}_{\text{time}}\}, & \text{diff}_{\text{time}} \geq 0, \\ \infty, & \text{otherwise}, \end{cases} \tag{21}$$

where

$$\text{diff}_{\text{time}} = T_{\text{re}}^{(UAV_u)} - \frac{d(TASK_i, UAV_u) + d(TASK_i, o)}{v^{(UAV_u)}} - t_i^{\text{req}}, \tag{22}$$

and $d(\cdot)$ denotes Euclidean travel distance, *o* is the return base, and $v^{(UAV_u)}$ is UAV speed. Similarly, the computation margin penalty is:

$$\Delta_{\text{comp}} = \begin{cases} \exp\{-c_{\text{comp}} \cdot \text{diff}_{\text{comp}}\}, & \text{diff}_{\text{comp}} \geq 0, \\ \infty, & \text{otherwise}, \end{cases} \tag{23}$$

with

$$\text{diff}_{\text{comp}} = C_{\text{re}}^{(UAV_u)} - c_i^{\text{req}}. \tag{24}$$

A negative margin in either dimension renders the assignment infeasible.

Neighborhood update mechanism
After the best UAV u^* is identified for task $TASK_i$, the SOM updates not only the winning UAV's feature vector but also those of its neighbors in the capability-topology space:

$$n_{u,u^*} = \begin{cases} 1, & u = u^*, \\ \exp\left(-\frac{S_{u,u^*}}{c_s}\right), & u \neq u^*,\ |\psi_{UAV_u} - \psi_{UAV_{u^*}}| \leq 1, \\ 0, & \text{otherwise}, \end{cases} \tag{25}$$

where S_{u,u^*} is the SOM grid distance. The feature update rule is:

$$\text{INFO}_{UAV_u}(r+1) = \text{INFO}_{UAV_u}(r) + n_{u,u^*}\left[\text{INFO}_{TASK_i} - \text{INFO}_{UAV_u}(r)\right], \tag{26}$$

which moves UAV feature vectors toward the matched task, facilitating adaptive clustering in subsequent iterations.

Algorithm flow
The complete pre-assignment and re-assignment procedure is summarized in Algorithm 1. Initially, all feasible UAV-task pairs are identified based on capability compatibility. The SOM then iteratively adjusts UAV feature vectors to minimize the matching distance in (17), with neighborhood updates ensuring topological consistency. After *R* iterations, each task is bound to the UAV with the smallest feasible matching distance, and UAV resources are updated to reflect the assigned workload.

```
Input: Task vectors $\text{INFO}_{TASK_i}$ for $i = 1, \ldots, M$; UAV vectors $\text{INFO}_{UAV_u}$ for $u = 1, \ldots, U$
Output: Matching set $\mathscr{M} = \{(i, u^*)\}$
$\mathscr{M} \leftarrow \emptyset$;
for $i \leftarrow 1$ to $M$ do
    $\mathscr{U}_{\text{eligible}} \leftarrow \{u \mid |\psi_{UAV_u} - \phi_{TASK_i}| \leq 1\}$;
    for $r \leftarrow 1$ to $R$ do
        for $u \in \mathscr{U}_{\text{eligible}}$ do
            Compute $D^{(i,u)}_{\text{INFO}}$ via Eqs. (17)–(24);
        $u^* \leftarrow \arg\min_{u \in \mathscr{U}_{\text{eligible}}} D^{(i,u)}_{\text{INFO}}$;
        if $r < R$ then
            for $u \in \mathscr{U}_{\text{eligible}}$ do
                Update $\text{INFO}_{UAV_u}$ using Eq. (26);
    Update $\text{INFO}_{UAV_{u^*}}$ by replacing position/type with task attributes and deducting the corresponding consumed
      resources;
    $\mathscr{M} \leftarrow \mathscr{M} \cup \{(i, u^*)\}$;
```

Algorithm 1. Pre-assignment and re-assignment algorithm (PR) for UAV remote sensing.

RL-based task sequence adjustment algorithm for UAV remote sensing

In UAV-assisted remote sensing missions, task allocation is often determined through pre-assignment or re-assignment stages. However, after allocation, the actual execution order of tasks critically affects overall mission efficiency. This is because UAVs operate in dynamic environments where inter-task distances, resource availability, and task priorities can change during flight. For example, after completing a given task $TASK_i$, the UAV must select its next task from the remaining set $\{TASK_j \mid j \neq i\}$; the choice will lead to different travel distances, energy consumption, and computational loads, thus impacting mission completion time and success rate. These variations essentially correspond to different state transitions and reward outcomes in a sequential decision-making process.

Importantly, the TSA module operates only on the small, feasibility-filtered task subset assigned to each UAV, because the preceding DMMP and PR stages have already removed infeasible or resource-incompatible tasks. This substantially reduces the dimensionality of the decision space and makes lightweight RL methods practical and efficient for real-time execution.

To adapt to such dynamics and achieve near-optimal task sequences under heterogeneous spatial, temporal, and resource constraints, the task sequence adjustment problem is formulated as a Markov Decision Process (MDP). RL enables UAVs to iteratively interact with the environment, evaluate the impact of different execution orders, and learn policies that maximize cumulative mission rewards. This approach effectively updates UAV flight paths in real time while balancing efficiency, priority satisfaction, and resource sustainability. In this work, the novelty of TSA lies in the resource-aware and feasibility-preserving MDP formulation rather than the specific choice of RL algorithm. Q-learning is selected due to its stability, interpretability, and suitability for onboard real-time decision making within the compact state–action spaces generated by the upstream modules.

State and action space definition

Let the assigned task sequence for UAV u be $\{TASK_1, TASK_2, \ldots, TASK_m\}$. The state space is defined as:

$$S = \{s_0, s_1, \ldots, s_m, s_{m+1}\}, \tag{27}$$

where:

- $s_j = (X_j, Y_j, pri_j) \in \mathbb{R}^3$ denotes the spatial coordinates (X_j, Y_j) and priority pri_j of $TASK_j$.
- s_0 denotes the UAV's current position and resource status after pre-assignment or re-assignment.
- s_{m+1} is the terminal state representing the return to the base station.

The action space is:

$$A = \{a_j \mid a_j : s_j \to s_{j'},\ j' \neq j\}, \tag{28}$$

where action a_j represents transitioning from the current task $TASK_j$ to the next selected task $TASK_{j'}$ in the sequence. Because the action space is restricted to the UAV's assigned tasks and excludes infeasible transitions filtered by the PR stage, the TSA module avoids the exponential explosion common in global routing problems.

Reward function with residual time and computing constraints

To jointly optimize spatial efficiency, task urgency, and UAV resource sustainability, the reward function is formulated as:

$$R(s_j, a_j) = \frac{c_d}{d(TASK_j, TASK_{j'})} + c_p \cdot pri_{j'} + c_t \cdot \frac{T_{\text{re}}^{(UAV_u)}}{T_{\max}^{(UAV_u)}} + c_c \cdot \frac{C_{\text{re}}^{(UAV_u)} - C_{TASK_{j'}}}{C_{\max}^{(UAV_u)}}, \tag{29}$$

where:

- $d(TASK_j, TASK_{j'})$ is the Euclidean distance between tasks *j* and j'.
- $pri_{j'}$ is the priority of $TASK_{j'}$.
- $T_{\text{re}}^{(UAV_u)}$ is the residual flight time before reaching the endurance limit $T_{\max}^{(UAV_u)}$.
- $C_{\text{re}}^{(UAV_u)}$ is the residual computational capacity of UAV *u*.
- $C_{TASK_{j'}}$ is the computational requirement of $TASK_{j'}$.
- $C_{\max}^{(UAV_u)}$ is the maximum computational capacity of UAV *u*.
- c_d, c_p, c_t, and c_c are positive weighting coefficients for distance minimization, priority maximization, endurance preservation, and computational resource sufficiency, respectively.

The first two terms encourage UAVs to execute spatially closer and higher-priority tasks. The third term favors UAVs with higher residual endurance for longer travel segments, and the fourth term prioritizes UAVs with sufficient computing capability for processing-intensive remote sensing tasks. By jointly encoding spatial cost, task importance, residual endurance, and computation feasibility, the reward function captures the multi-resource coupling unique to heterogeneous sensing-processing missions. This design constitutes the core innovation of TSA, enabling dynamic feasibility-preserving decision making beyond standard Q-learning applications.

The Q-value update follows the standard temporal-difference rule:

$$Q[S, A] \leftarrow (1 - \alpha)Q[S, A] + \alpha \left(R(S, A) + \gamma \max_{A'} Q[S', A'] \right), \tag{30}$$

where α is the learning rate, and γ is the discount factor.

Task sequence adjustment algorithm

```
Input  : Assigned task set $\{TASK_1, TASK_2, \ldots, TASK_m\}$ for UAV $u$ with positions $(X, Y)$, priorities $pri$, and
         computational requirements $C_{TASK}$; UAV's current state $(X_u, Y_u, T_{\text{re}}^{(UAV_u)}, C_{\text{re}}^{(UAV_u)})$.
Output : Optimized execution sequence $\{S_j^*\}$.
Define state space $S$ as in (27) and action space $A$ as in (28)
Compute rewards $R(s, a)$ using (29) and initialize the Q-table $Q[S, A]$
for each episode do
    $s \leftarrow s_0$                                   // initial UAV state after pre-/re-assignment
    while $s \neq s_{m+1}$ do
        Select action $a$ via $\varepsilon$-greedy policy
        Execute $a$ (transition from $TASK_j$ to $TASK_{j'}$); observe $R(s, a)$
        Update $Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha(R(s, a) + \gamma \max_{a'} Q(s', a'))$
        $s \leftarrow s'$
return $\{S_j^*\}$ from the final Q-table
```

Algorithm 2. Task sequence adjustment algorithm (TSA).

After initial task allocation, the UAV needs to dynamically determine the optimal execution order of the remaining tasks, taking into account spatial distances, task priorities, residual flight time, and computational capacity. This process is modeled as an iterative decision-making problem in which the UAV interacts with the environment, evaluates alternative next-task choices, and updates its decision policy based on observed rewards. The proposed Task Sequence Adjustment (TSA) algorithm leverages Q-learning to learn a near-optimal policy through repeated simulations or online operation.

The TSA algorithm operates by continuously evaluating the trade-offs between travel efficiency and mission priority. At each decision step, the UAV selects the next task that maximizes its long-term cumulative reward, rather than only minimizing the immediate travel cost. This approach allows the UAV to adapt to changing environmental conditions and heterogeneous task demands, while maintaining energy and computational feasibility for the remainder of the mission. Furthermore, the compact and structured decision space produced by earlier DMMP and PR stages enables Q-learning to converge rapidly, making it highly suitable for onboard execution without requiring computationally expensive deep RL algorithms.

Parameter	Value	Description
$v^{(UAV_u)}$	$20\,\mathrm{m/s}$	UAV cruising speed
c_ϕ	3	Task type matching coefficient
c_{RES}	6	Resource matching coefficient
c_s	20	Node influence coefficient
c_d	6	Distance penalty coefficient
c_p	1	Priority weighting coefficient
$T_{\max}^{(UAV_u)}$ $(\psi_{UAV_u} = -1)$	1800 s	Max flight time (Type -1 UAV)
$T_{\max}^{(UAV_u)}$ $(\psi_{UAV_u} = 0)$	1500 s	Max flight time (Type 0 UAV)
$T_{\max}^{(UAV_u)}$ $(\psi_{UAV_u} = 1)$	1200 s	Max flight time (Type 1 UAV)
$C_{\max}^{(UAV_u)}$ $(\psi_{UAV_u} = -1)$	0 GHz·s	Max computational capacity (Type -1 UAV)
$C_{\max}^{(UAV_u)}$ $(\psi_{UAV_u} = 0)$	60 GHz·s	Max computational capacity (Type 0 UAV)
$C_{\max}^{(UAV_u)}$ $(\psi_{UAV_u} = 1)$	40 GHz·s	Max computational capacity (Type 1 UAV)

Table 1. Basic simulation parameter settings.

UAV	Position POS		Type	Remaining flight time *T*	Remaining capacity *C*
ID	*X*	*Y*	ψ_{UAV_u}	(s)	(GHz s)
0	3.24	5.10	1	1200	40.0
1	4.62	1.14	0	1500	60.0
2	5.11	7.43	-1	1800	0.0

Table 2. Initial UAV parameter configuration.

Task	Position POS		Type	Required time T_{TASK}	Required capacity C_{TASK}
ID	*X*	*Y*	ϕ_{TASK_i}	(s)	(GHz s)
0	7.24	1.23	-1	42	0.0
1	1.86	0.94	-1	103	17.73
2	6.12	1.84	1	99	16.16
3	2.81	3.84	1	96	14.95
4	6.29	7.97	-1	98	0.0
5	6.89	2.80	0	90	10.45
6	7.28	6.99	-1	55	0.0
7	3.02	1.10	-1	72	0.0
8	7.43	1.52	1	92	15.66

Table 3. Initial task node parameter configuration.

Experimental setup and performance evaluation

Experimental environment setup

In this section, simulation experiments are conducted to evaluate the performance of the proposed **DMMP-PR-TSA** algorithm in UAV-based remote sensing scenarios with edge computing. The evaluation focuses on two aspects: (1) *Matching effectiveness*: The PR stage, implemented via a Self-Organizing Map (SOM) network, is tested to verify the efficiency of pre-assignment and re-assignment of remote sensing tasks; (2) *Completion rate improvement*: The TSA stage, based on reinforcement learning (RL), is compared with baseline metaheuristic algorithms including PSO and DPSO to demonstrate the advantage of dynamic task sequence adjustment.

The basic simulation parameters are listed in Table 1. The initial UAV configuration and task node configuration, including spatial position, type, and resource attributes, are shown in Tables 2 and 3, respectively.

Pre-assignment and re-assignment in UAV-based remote sensing

To evaluate the adaptability and efficiency of the proposed DMMP-PR-TSA algorithm in multi-task UAV-based remote sensing scenarios, we investigate both the task pre-assignment and task re-assignment phases. The

UAV ID	PR assignment	PSO/DPSO assignment
0	$\{TASK_1, TASK_3, TASK_8\}$	$\{TASK_1, TASK_2, TASK_8\}$
1	$\{TASK_0, TASK_2, TASK_5, TASK_7\}$	$\{TASK_0, TASK_3, TASK_6\}$
2	$\{TASK_4, TASK_6\}$	$\{TASK_4, TASK_5, TASK_7\}$

Table 4. comparison of task pre-assignment between PR and PSO/DPSO.

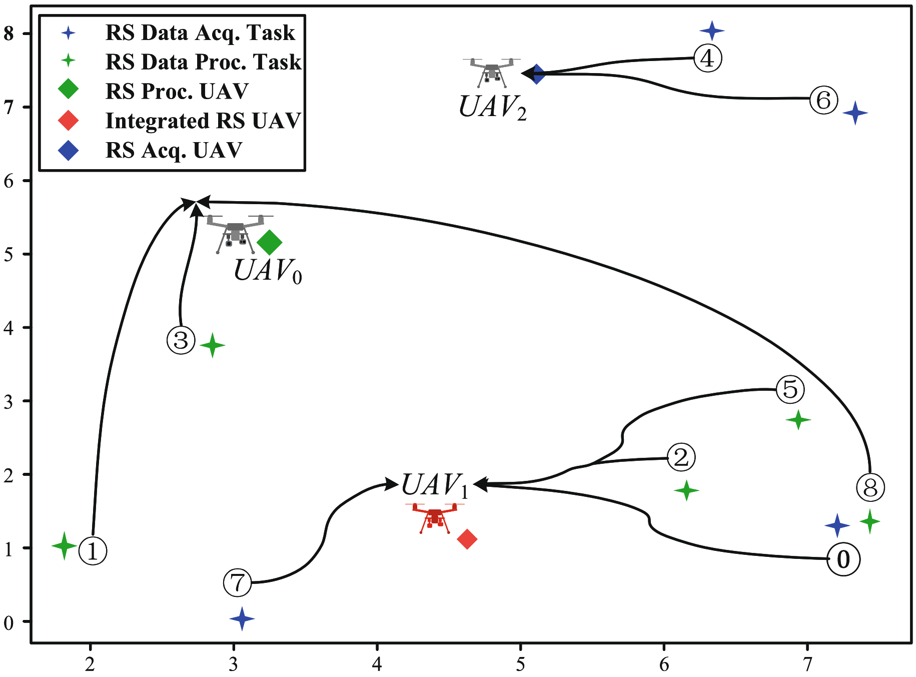


Fig. 5. Task pre-assignment results using the proposed PR algorithm.

experimental setting emulates representative mission profiles such as post-disaster assessment, environmental monitoring, and infrastructure inspection, where UAVs must accomplish remote sensing tasks under strict constraints on onboard computational resources and flight endurance. The operational extent is derived directly from the georeferenced HTCD dataset covering the Chişinău urban area [45]. The dataset contains one satellite image (11K × 15K pixels; ground resolution 0.5971 m) and 15 UAV image tiles (in total 1.38M × 1.04M pixels; ground resolution 7.465 cm), provided as GeoTIFFs and co-registered by manually selected control points with a polynomial model. We rasterize this extent and compute a cell-wise task-intensity score from the dataset's pixel-level urban change labels (buildings, roads, and other man-made features), which serves as a proxy for sensing demand and image-processing workload. Based on these scores and UAV heterogeneity (residual flight time and onboard computing capacity), we perform *capacity-constrained power-diagram partitioning* with local refinement (as described in our method section) to obtain contiguous working subregions whose aggregated expected workload matches each UAV's capacity, thereby balancing coverage and reducing inter-region traversal. These subregions provide the basis for pre-assignment and re-assignment. Task computing demands are expressed in GHz · s to reflect typical image-processing workloads in UAV-based sensing applications.

Task pre-assignment

Table 4 compares the task pre-assignment results of the proposed PR Algorithm with those of the Particle Swarm Optimization (PSO) and Discrete Particle Swarm Optimization (DPSO) baselines. The PR algorithm is designed to jointly consider spatial proximity and residual computational resources during allocation, thereby aligning task requirements with UAV capabilities in heterogeneous remote sensing scenarios. In contrast, the PSO algorithm is a population-based metaheuristic that optimizes candidate solutions by simulating the collective behavior of particle swarms, exhibiting strong global search capabilities but lacking explicit resource-awareness. DPSO extends PSO to discrete solution spaces, which allows it to address combinatorial allocation problems such as UAV task scheduling; however, it still inherits the same resource-matching limitations.

Experimental results show that, across all tested scenarios, the PR algorithm consistently assigns each task to a UAV that satisfies both the task-type compatibility constraint ϕ and the residual computational capacity requirement W, thus improving deadline compliance and mission reliability. By comparison, PSO and DPSO may allocate computation-intensive tasks to UAVs with insufficient remaining processing capacity, which

Task ID	X (km)	Y (km)	ϕ	T (s)	W (GHz · s)
9	0.70	1.62	− 1	52	0
10	7.34	0.60	1	79	12.37
11	7.20	4.76	− 1	55	0
12	3.19	7.22	− 1	63	0
13	7.30	3.56	− 1	56	0
14	4.52	6.08	− 1	65	0
15	1.00	2.83	− 1	41	0
16	6.20	3.30	− 1	69	0
17	7.26	0.74	1	86	13.37

Table 5. Parameters of newly inserted tasks.

UAV ID	PR assignment	PSO/DPSO assignment
0	$\{\mathrm{TASK}_1, \mathrm{TASK}_3, \mathrm{TASK}_8, \mathrm{TASK}_{10}, \mathrm{TASK}_{17}\}$	$\{\mathrm{TASK}_1, \mathrm{TASK}_3, \mathrm{TASK}_8, \mathrm{TASK}_{10}\}$
1	$\{\mathrm{TASK}_0, \mathrm{TASK}_2, \mathrm{TASK}_5, \mathrm{TASK}_7, \mathrm{TASK}_{15}\}$	$\{\mathrm{TASK}_0, \mathrm{TASK}_2, \mathrm{TASK}_5, \mathrm{TASK}_{15}\}$
2	$\{\mathrm{TASK}_4, \mathrm{TASK}_6, \mathrm{TASK}_{11}, \mathrm{TASK}_{12}, \mathrm{TASK}_{13}, \mathrm{TASK}_{14}, \mathrm{TASK}_{16}\}$	$\{\mathrm{TASK}_4, \mathrm{TASK}_6, \mathrm{TASK}_{11}, \mathrm{TASK}_{12}, \mathrm{TASK}_{13}, \mathrm{TASK}_{14}, \mathrm{TASK}_{16}\}$

Table 6. Comparison of task re-assignment (new task insertion) between PR and PSO/DPSO.

increases the likelihood of deadline violations and leads to less efficient utilization of heterogeneous onboard resources.

For instance, in Fig. 5, $TASK_8$, which requires high-resolution image processing and thus demands high computational throughput, is assigned to UAV$_0$ by the PR strategy. This decision leverages UAV$_0$'s relatively higher residual resources compared with UAV$_1$, thereby lowering the probability of deadline violations. In contrast, the PSO/DPSO-based pre-assignment mixes $TASK_8$ with other tasks in a way that disregards residual capacity, resulting in more fragmented load distribution and potentially prolonging the overall mission completion time.

Task re-assignment

New task insertion: In realistic UAV-based remote sensing missions, dynamic environmental changes or emergent observation demands often necessitate the insertion of new tasks into an ongoing mission plan. To emulate such conditions, nine additional sensing tasks are introduced after the initial pre-assignment stage. These tasks represent urgent requests such as post-disaster site mapping, rapid surveillance of emerging hotspots, or detection of sudden environmental anomalies. The parameters of the newly inserted tasks are summarized in Table 5, where (X, Y) specifies the geographic location, ϕ denotes the task type ($\phi = 1$ for computation-intensive data processing tasks and $\phi = -1$ for acquisition-only tasks), T indicates the required execution time, and C corresponds to the onboard computing workload in $\mathrm{GHz} \cdot \mathrm{s}$, reflecting typical UAV-based image processing demands.

Table 6 compares the re-assignment results for the new task insertion scenario. The proposed PR algorithm leverages both spatial proximity and residual computational capacity, ensuring that high-demand tasks (e.g., $TASK_{10}$ and $TASK_{17}$) are assigned to UAVs capable of processing them without resource saturation. This allows the system to absorb newly arrived tasks while preserving feasibility margins on each UAV. In contrast, PSO and DPSO sometimes assign computation-intensive tasks to UAVs with limited residual resources, thereby increasing the risk of missed deadlines and compromising mission efficiency.

Figure 6 visualizes the spatial allocation after re-assignment. Markers differentiate acquisition-only and processing-capable UAVs, while task symbols indicate processing requirements. Under the PR strategy, the resulting allocation exhibits geographically coherent clusters and a more balanced distribution of processing workloads, which jointly reduce unnecessary detours and improve robustness against sudden task surges.

Task location update: In UAV remote sensing, moving targets or evolving observation areas can cause previously assigned task coordinates to become outdated. To simulate this, the location of $TASK_0$ is updated to $(7.24,\ 6.95)$.

As shown in Table 7 and Fig. 7, the PR algorithm reassigns $TASK_0$ to the UAV with closer proximity and sufficient residual computational resources (UAV$_2$), thereby reducing travel time and avoiding overload. The baselines, in contrast, tend to keep the original assignment even after the location shift, which yields longer flight paths and tighter resource margins. This adaptive reallocation is critical in time-sensitive sensing missions where target drift occurs.

UAV failure scenario Another common challenge in UAV-based remote sensing is sudden platform failure, which may occur due to mechanical malfunction, communication breakdown, or energy depletion. In this scenario, UAV$_1$ fails immediately after completing $TASK_0$, and thus only UAV$_0$ and UAV$_2$ remain operational. The PR algorithm redistributes UAV$_1$'s remaining tasks to the surviving UAVs with minimal added travel distance and sufficient residual computational capacity. As presented in Table 8 and Fig. 8, PR yields a spatially coherent

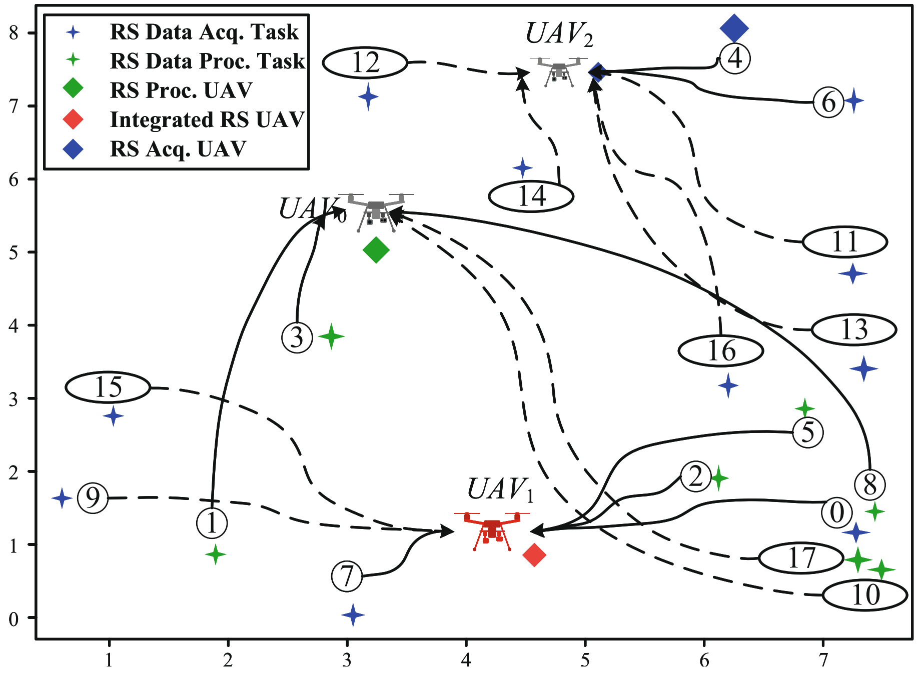


Fig. 6. Task re-assignment results for the new task insertion scenario.

UAV ID	PR assignment	PSO/DPSO assignment
0	$\{\text{TASK}_1, \text{TASK}_3, \text{TASK}_8\}$	$\{\text{TASK}_1, \text{TASK}_2, \text{TASK}_8\}$
1	$\{\text{TASK}_2, \text{TASK}_5, \text{TASK}_7\}$	$\{\text{TASK}_0, \text{TASK}_3, \text{TASK}_6\}$
2	$\{\text{TASK}_0, \text{TASK}_4, \text{TASK}_6\}$	$\{\text{TASK}_4, \text{TASK}_5, \text{TASK}_7\}$

Table 7. Comparison of task re-assignment (location update) between PR and PSO/DPSO.

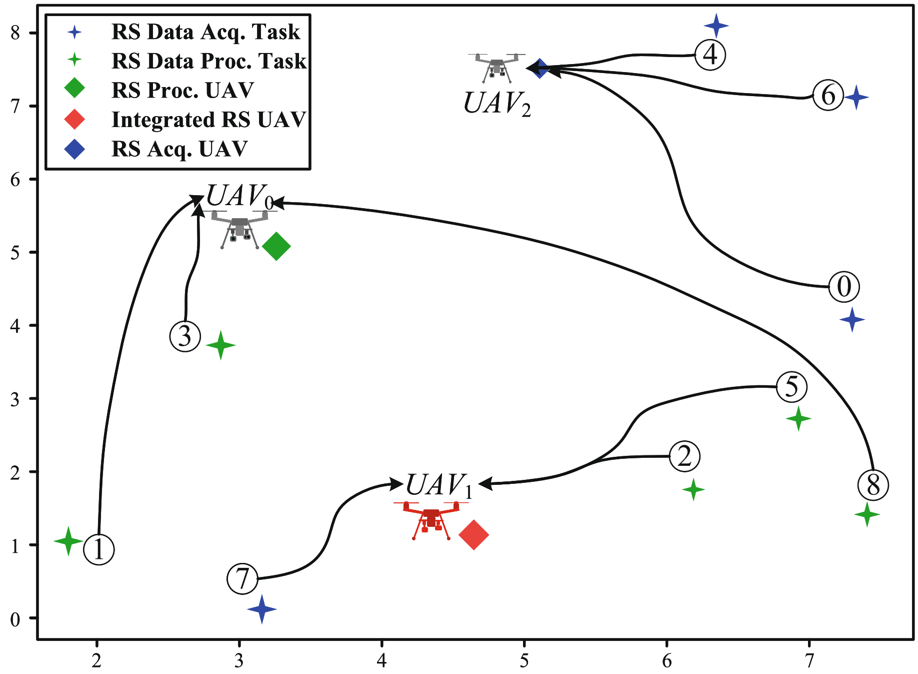


Fig. 7. Task reallocation after a task-location update using the PR algorithm.

UAV ID	PR assignment	PSO/DPSO assignment
0	{TASK$_1$, TASK$_2$, TASK$_3$, TASK$_5$, TASK$_8$}	{TASK$_1$, TASK$_2$, TASK$_3$, TASK$_8$}
2	{TASK$_4$, TASK$_6$, TASK$_7$}	{TASK$_4$, TASK$_5$, TASK$_7$}

Table 8. Comparison of task re-assignment (UAV failure) between PR and PSO/DPSO.

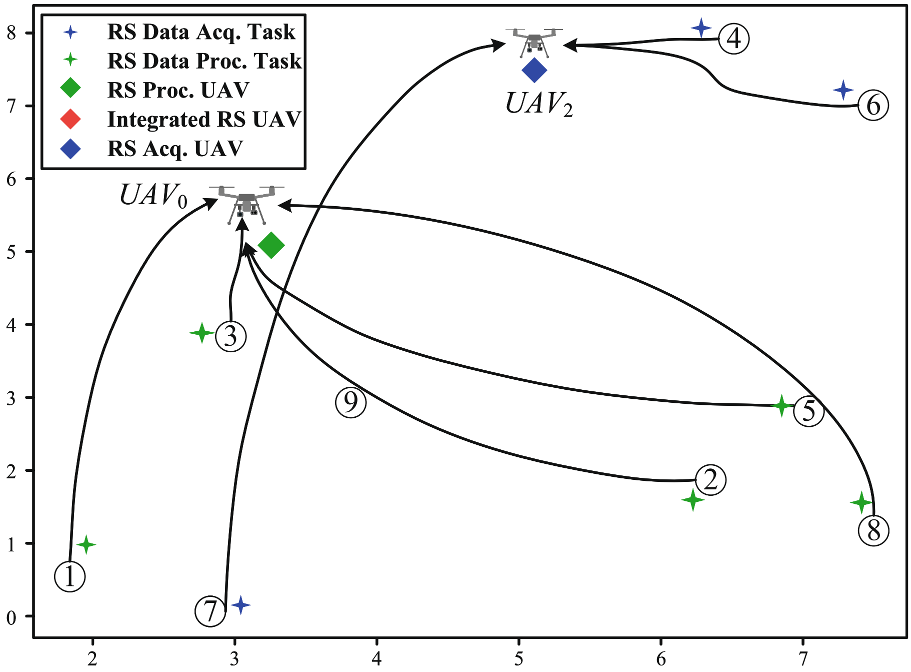


Fig. 8. Task reallocation after UAV$_1$ failure using the PR algorithm. Numbered circles (1–8) mark the task indices after $TASK_0$ is completed; only UAV$_0$ and UAV$_2$ are active. Solid curves show PR routes: UAV$_0$ executes {1,2,3,5,8} and UAV$_2$ executes {4,6,7}.

Task ID	Priority level	Deadline (s)
0	1	1200
1	3	800
2	3	800
3	1	1200
4	2	1000
5	3	800
6	1	1200
7	2	1000
8	3	800

Table 9. Initial priority levels and deadlines of tasks.

reassignment and avoids overload, whereas PSO/DPSO may cause uneven clustering around certain regions. This demonstrates that the proposed re-assignment mechanism can maintain service continuity and balanced utilization even under single-UAV failures, which is essential for safety-critical remote sensing operations.

Dynamic task sequence adjustment based on RL

In the proposed RL-based task sequence adjustment model, the reward function is determined by both the spatial coordinates of each task (critical for path optimization in remote sensing scenarios) and its assigned priority level, reflecting the operational urgency and time-sensitivity inherent in UAV-based remote sensing missions such as disaster assessment, environmental monitoring, and infrastructure inspection. For the initial task set, three priority levels are defined, as summarized in Table 9: priority levels 1, 2, and 3 correspond to maximum allowable completion times (deadlines) of 1200 s, 1000 s, and 800 s, respectively. For newly inserted tasks, the corresponding deadlines are 1500 s, 1250 s, and 1000 s, providing additional scheduling flexibility during in-

UAV ID	TSA	PSO/DPSO
0	$\tau(TASK_3, TASK_1, TASK_8)$	$\tau(TASK_8, TASK_1, \mathbf{TASK_2})$
1	$\tau(TASK_2, TASK_5, TASK_0, TASK_7)$	$\tau(TASK_6, TASK_3, TASK_0, \mathbf{TASK_5})$
2	$\tau(TASK_6, TASK_4)$	$\tau(\mathbf{TASK_5}, TASK_7, TASK_4)$
Completion rate	100%	77.78%

Table 10. Comparison of task-sequence configuration between TSA and PSO/DPSO.

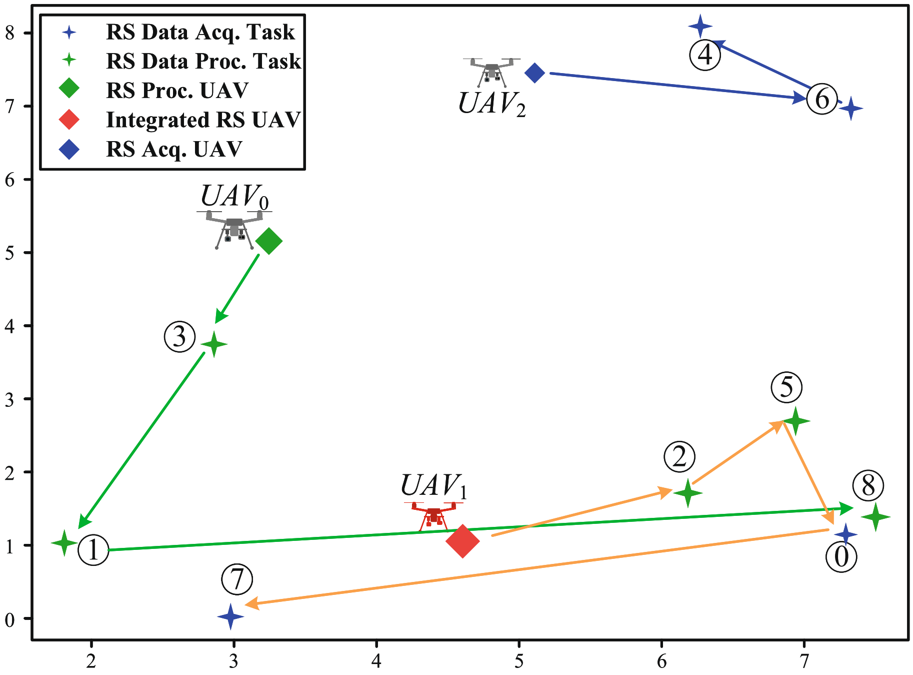


Fig. 9. TSA-based task-sequence configuration. Colored polylines correspond to $UAV_0/UAV_1/UAV_2$; numbered markers denote task IDs (0–8). The polyline order depicts the execution sequence, showing that TSA schedules UAV_0 for $\{3, 1, 8\}$, UAV_1 for $\{2, 5, 0, 7\}$, and UAV_2 for $\{6, 4\}$.

Task point	9	10	11	12	13	14	15	16	17
Priority	2	2	2	2	3	2	1	1	3
Deadline ($\leq$)	1250 s	1250 s	1250 s	1250 s	1000 s	1250 s	1500 s	1500 s	1000 s

Table 11. Priority and deadline settings for new task points.

flight operations while preventing overload of onboard computing resources required for data processing and transmission.

Initial task sequence configuration

Table 10 compares the task execution sequences produced by the proposed TSA (Task Sequence Adjustment) and the PSO/DPSO baselines under identical pre-assignment results. As illustrated in Fig. 9, TSA generates spatially coherent routes and well-ordered execution sequences, achieving a $\mathbf{100}\%$ on-time completion rate when task density is moderate. In contrast, PSO/DPSO achieves only $\mathbf{77.78}\%$, mainly because it (i) places high-priority tasks near the end of a sequence (e.g., $\mathbf{TASK_2}$ for UAV_0) and (ii) assigns data-intensive tasks to UAVs with limited processing capability (e.g., $\mathbf{TASK_5}$ to UAV_2), which leads to deadline violations or infeasible schedules.

Task sequence updates after reallocation

New task insertion To emulate real-time mission changes commonly encountered in dynamic UAV-based remote sensing scenarios, nine new tasks are inserted into the mission schedule (Table 11). Each new task is assigned a priority and deadline that reflect the urgency of unexpected events such as sudden environmental

UAV	Task sequence update status		
	TSA	PSO	DPSO
0	$\tau(\mathrm{TASK}_3, \mathrm{TASK}_1, \mathrm{TASK}_{17}, \mathrm{TASK}_8, \mathrm{TASK}_{10})$	$\tau(\mathrm{TASK}_{17}, \mathrm{TASK}_5, \mathrm{TASK}_8, \mathrm{TASK}_3, \mathbf{TASK_1})$	$\tau(\mathrm{TASK}_8, \mathrm{TASK}_1, \mathbf{TASK_2})$
1	$\tau(\mathrm{TASK}_2, \mathrm{TASK}_5, \mathrm{TASK}_7, \mathrm{TASK}_9, \mathrm{TASK}_{15}, \mathrm{TASK}_0)$	$\tau(\mathrm{TASK}_{11}, \mathrm{TASK}_9, \mathrm{TASK}_{12}, \mathrm{TASK}_{10}, \mathrm{TASK}_4, \mathrm{TASK}_2)$	$\tau(\mathrm{TASK}_6, \mathrm{TASK}_{12}, \mathrm{TASK}_3, \mathrm{TASK}_{17}, \mathrm{TASK}_9, \mathrm{TASK}_{13}, \mathrm{TASK}_{16}, \mathrm{TASK}_0)$
2	$\tau(\mathrm{TASK}_6, \mathrm{TASK}_{14}, \mathrm{TASK}_{12}, \mathrm{TASK}_{11}, \mathrm{TASK}_{13}, \mathrm{TASK}_{16}, \mathrm{TASK}_4)$	$\tau(\mathrm{TASK}_6, \mathrm{TASK}_{14}, \mathrm{TASK}_{15}, \mathrm{TASK}_{16}, \mathrm{TASK}_{13}, \mathbf{TASK_7}, \mathrm{TASK}_0)$	$\tau(\mathbf{TASK_5}, \mathrm{TASK}_{14}, \mathrm{TASK}_{11}, \mathrm{TASK}_{15}, \mathrm{TASK}_7, \mathrm{TASK}_4, \mathbf{TASK_{10}})$
Completion rate	88.89%	72.22%	77.78%

Table 12. Comparison of task sequence update results using TSA algorithm and PSO/DPSO algorithms (new tasks).

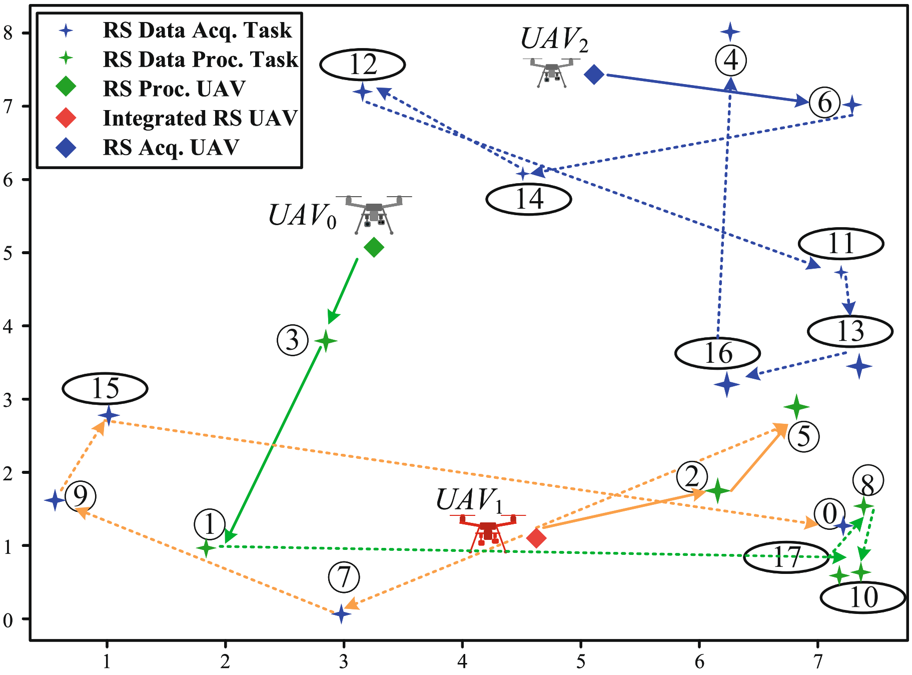


Fig. 10. TSA-based task sequence update results after inserting new remote sensing tasks. The colored paths correspond to UAV$_0$, UAV$_1$, and UAV$_2$, with numbered markers indicating task IDs. The execution orders match the TSA sequences in Table 12.

UAV ID	TSA	PSO/DPSO
0	$\tau(TASK_3, TASK_1, TASK_8)$	$\tau(TASK_8, TASK_1, \mathbf{TASK_2})$
1	$\tau(TASK_2, TASK_5, TASK_7)$	$\tau(\mathbf{TASK_5}, TASK_3, TASK_0)$
2	$\tau(TASK_6, TASK_0, TASK_4)$	$\tau(\mathbf{TASK_5}, TASK_7, TASK_4)$
Completion rate	100%	77.78%

Table 13. Task sequence update (location update).

anomalies or post-disaster monitoring requirements. Priorities are mapped to maximum allowable completion times to ensure that more urgent tasks are executed earlier. Specifically, priority levels 1, 2, and 3 correspond to deadlines of 1500 s, 1250 s, and 1000 s, respectively, for the newly inserted tasks.

UAV ID	TSA	PSO/DPSO
0	$\tau(TASK_3, TASK_2, TASK_5, TASK_8, \mathbf{TASK_1})$	$\tau(TASK_2, TASK_1, TASK_3, TASK_8)$
2	$\tau(TASK_4, TASK_6, TASK_7)$	$\tau(TASK_4, \mathbf{TASK_5}, TASK_7)$
Completion rate	88.89%	66.67%

Table 14. Task sequence update (UAV failure).

UAV ID	TSA	PSO/DPSO
0	$\tau(TASK_3, TASK_8)$	$\tau(TASK_8)$
1	$\tau(TASK_7, TASK_5, TASK_0)$	$\tau(TASK_3, TASK_0)$
2	$\tau(TASK_6, TASK_4)$	$\tau(TASK_5, TASK_4)$
Completion rate	100%	71.43%

Table 15. Task sequence update (task cancellation).

The proposed TSA method dynamically reorders task sequences to minimize deadline violations while considering the UAVs' residual computing resources. This results in an on-time completion rate of 88.89%, compared to 72.22% for PSO and 77.78% for DPSO (Table 12 and Fig. 10). In the PSO and DPSO sequences, several high-priority tasks (e.g., TASK_1, TASK_2, and TASK_5) are positioned near the end of the sequence or assigned to UAVs with limited processing capabilities, increasing the risk of deadline violations and reducing the overall completion rate.

Task location update When the location of TASK_0 is updated to $(7.24,\ 4.15)$—simulating target position refinement common in remote sensing—the TSA method reorders the task sequences considering both spatial proximity and residual UAV computing capacity for data processing, achieving 100% completion versus 77.78% for PSO/DPSO (Table 13).

UAV failure If UAV_1 experiences a failure after completing TASK_0—a critical scenario in remote sensing missions requiring operational robustness—after PR-based task reassignment, TSA reorders the sequences for the remaining UAVs to minimize detour and respect computing constraints, achieving 88.89% completion versus 66.67% for PSO/DPSO (Table 14).

Task cancellation If TASK_1 and TASK_2 are cancelled during execution—mimicking situations where remote sensing targets become inaccessible—TSA promptly reorders the task sequences while maintaining optimal use of UAV computing resources, ensuring 100% completion compared to 71.43% for PSO/DPSO (Table 15).

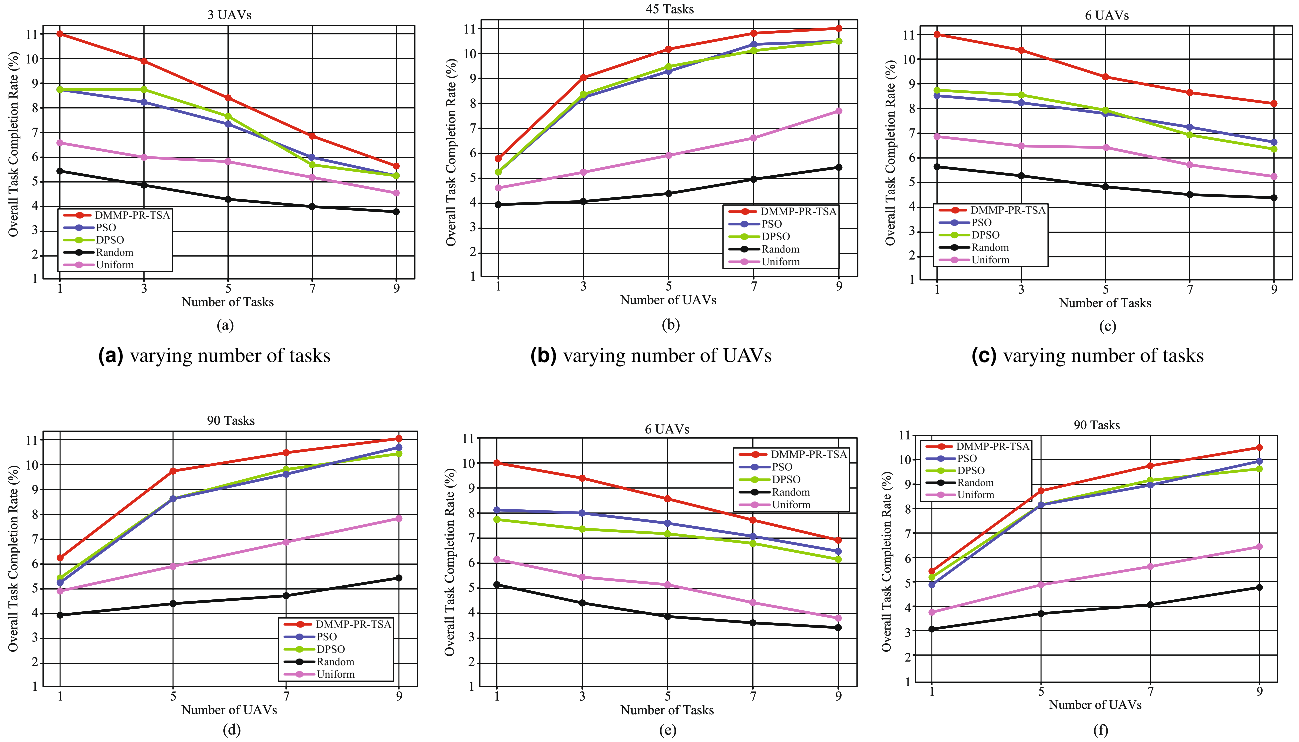
Advantages of the DMMP-PR-TSA algorithm

To validate the effectiveness of the proposed DMMP-PR-TSA algorithm in UAV remote sensing missions, we compare it against four baseline methods: Particle Swarm Optimization (PSO), Dynamic PSO (DPSO), Random assignment, and Uniform allocation. PSO and DPSO are meta-heuristic scheduling approaches that optimize task allocation based on particle search and dynamic parameter adaptation, respectively. Random assignment allocates tasks without optimization, while Uniform allocation evenly distributes tasks among UAVs without considering spatial or computational heterogeneity.

The experiments consider mixed workloads of UAV remote sensing data acquisition tasks and edge real-time processing tasks, with a fixed task-type ratio of 1:2. Pre-assignment and re-assignment are balanced at a 1:1 ratio, and high-priority tasks account for 50% of the total. Three UAV state configurations are tested: (1) balanced states (1:1:1), (2) acquisition-oriented fleet (1:2:3), and (3) processing-oriented fleet (3:2:1). For each configuration, the number of tasks or UAVs is varied to evaluate scalability and adaptability.

Figure 11a,c,e show the overall task completion rate versus the number of tasks under different fleet configurations. As task volume increases from 20 to 100, DMMP-PR-TSA consistently outperforms all baselines, achieving approximately 15–20% higher completion rates than PSO/DPSO. This improvement is mainly due to the multi-stage design of DMMP-PR-TSA, in which capacity-constrained region partitioning, feasibility-aware task reallocation, and resource-aware sequence optimization jointly reduce infeasible assignments and avoid local congestion on energy or computing resources that cannot be explicitly handled by single-stage meta-heuristics. This advantage is more pronounced in large-scale missions such as regional environmental monitoring or wide-area disaster mapping, where efficient integration of spatial path planning and onboard processing is critical.

Figure 11b,d,f illustrate the impact of varying the number of UAVs for fixed workloads (45 or 90 tasks). DMMP-PR-TSA maintains a 10–30% improvement in completion rate over PSO/DPSO across all fleet sizes. The results indicate that the proposed framework can still redistribute tasks and adjust local execution sequences effectively when the fleet becomes more unbalanced or when some UAVs are relatively resource-constrained, whereas the baselines tend to over-utilize a subset of platforms. This robustness to fleet composition changes is essential for dynamic remote sensing scenarios, such as emergency response, where UAVs may need to withdraw for battery replacement or redeploy to higher-priority regions.



(a) varying number of tasks **(b)** varying number of UAVs **(c)** varying number of tasks

(d) varying number of UAVs **(e)** varying number of tasks **(f)** varying number of UAVs

Fig. 11. Performance comparison of DMMP-PR-TSA with baseline algorithms under different UAV fleet configurations and task scales. (**a**,**c**,**e**) show completion rate versus the number of tasks; (**b**,**d**,**f**) show completion rate versus the number of UAVs.

Across all experiments, DMMP-PR-TSA consistently achieves higher completion rates for high-priority tasks by jointly optimizing spatial coverage for sensing tasks and computational resource allocation for processing tasks. These results confirm its scalability, adaptability, and priority-awareness, making it well-suited for time-sensitive, computation-intensive UAV remote sensing operations. From a computational perspective, the multi-stage DMMP-PR-TSA framework exhibits an overall time complexity that grows approximately linearly with the total number of tasks and UAVs under typical parameter settings, which is of the same order as PSO/DPSO while achieving noticeably higher task completion rates.

Training efficiency and adaptation: The TSA module is trained using lightweight tabular Q-learning on a per-UAV basis, where each UAV learns over its own feasibility-filtered task subset. Since the number of tasks per UAV is moderate due to the preceding partitioning and reallocation stages, the state—action space remains compact and convergence is typically achieved within a few hundred simulated mission episodes. This keeps the offline training cost comparable to meta-heuristic baselines, while the online decision-making phase is near-instantaneous.

Conclusions

This paper addresses the challenge of task scheduling for multi-heterogeneous UAVs in remote sensing operations, and proposes the DMMP-PR-TSA algorithm as a dynamic multi-stage mission planning framework for priority-aware, capacity-constrained scheduling of mixed sensing and edge-processing tasks. The PR algorithm based on SOM first accomplishes task pre-allocation and re-allocation triggered by dynamic task changes (such as addition, update, failure, cancellation) and dynamic regional adjustments. Then, through reinforcement learning, it dynamically adapts to the changes in the task sequence caused by regional variations and updates the flight trajectories. Experiments demonstrate that the algorithm effectively improves the overall task completion rate of UAV remote sensing tasks, providing a reliable technical solution for UAV remote sensing task scheduling in complex scenarios involving dynamic regions. At the same time, the current work mainly focuses on open or moderately obstructed environments, and the adaptability of the algorithm to extreme working conditions in highly cluttered urban scenes or complex obstacle-rich environments still has room for improvement. In subsequent research, we plan to extend the framework toward obstacle-aware region partitioning and urban-environment-oriented task planning, further refining the algorithm's robustness and practicality for real-world large-scale remote sensing missions.

Data availability

The dataset (comprising geo-referenced satellite/UAV image pairs and pixel-wise change labels) used in this study is the HTCD satellite–UAV heterogeneous change detection dataset released with SUNet and is publicly available at https://github.com/ShaoRuizhe/SUNet-change_detection.

Received: 28 August 2025; Accepted: 29 December 2025
Published online: 06 January 2026

References

1. Hao, H., Xu, C., Zhang, W., Yang, S. & Muntean, G.-M. Joint task offloading, resource allocation, and trajectory design for multi-UAV cooperative edge computing with task priority. *IEEE Trans. Mob. Comput.***23**, 8649–8663 (2024).
2. Tang, J. & Zeng, Y. UAV data acquisition and processing assisted by UGV-enabled mobile edge computing. *IEEE Trans. Ind. Inform.* (2025).
3. Al-Bakhrani, A. A., Li, M., Obaidat, M. S. & Amran, G. A. Moalf-uav-mec: Adaptive multi-objective optimization for UAV-assisted mobile edge computing in dynamic IOT environments. *IEEE Internet Things J.* (2025).
4. Tang, Y. et al. Integrated sensing, computation, and communication for UAV-assisted federated edge learning. *IEEE Trans. Wirel. Commun.* (2025).
5. Jia, Z. et al. Distributionally robust optimization for aerial multi-access edge computing via cooperation of UAVS and haps. *IEEE Trans. Mobile Comput.* (2025).
6. Li, J. et al. A learning-based stochastic game for energy efficient optimization of UAV trajectory and task offloading in space/aerial edge computing. *IEEE Trans. Veh. Technol.* (2025).
7. Cao, P. et al. UAV swarm cooperative search based on scalable multiagent deep reinforcement learning with digital twin-enabled sim-to-real transfer. *IEEE Trans. Mob. Comput.* (2025).
8. Wang, C. et al. Computing power in the sky: Digital twin-assisted collaborative computing with multi-UAV networks. *IEEE Trans. Veh. Technol.* (2025).
9. Teng, J., Sun, H., Liu, P. & Jiang, S. An improved transmvsnet algorithm for three-dimensional reconstruction in the unmanned aerial vehicle remote sensing domain. *Sensors***24**, 2064 (2024).
10. Dabiri, M. T., Hasna, M., Allhunibal, S. & Qaraq, K. Joint UAV-based directional thz communication and 3d map construction. In *2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall)*, 1–6 (IEEE, 2024).
11. Sohl, M. A. & Mahmood, S. A. Low-cost UAV in photogrammetric engineering and remote sensing: Georeferencing, dem accuracy, and geospatial analysis. *J. Geovis. Spat. Anal.***8**, 14 (2024).
12. Dritsas, E. & Trigka, M. Remote sensing and geospatial analysis in the big data era: A survey. *Remote Sens.***17**, 550 (2025).
13. Wan, P., Xu, G., Chen, J. & Zhou, Y. Deep reinforcement learning enabled multi-UAV scheduling for disaster data collection with time-varying value. *IEEE Trans. Intell. Transp. Syst.***25**, 6691–6702 (2024).
14. Ruess, S., Paulus, G. & Lang, S. Automated derivation of vine objects and ecosystem structures using UAS-based data acquisition, 3d point cloud analysis, and obia. *Appl. Sci.***14**, 3264 (2024).
15. Sedlak, B., Murturi, I., Donta, P. K. & Dustdar, S. A privacy enforcing framework for data streams on the edge. *IEEE Trans. Emerg. Top. Comput.***12**, 852–863. https://doi.org/10.1109/TETC.2023.3315131 (2024).
16. Dehury, C. K., Kumar Donta, P., Dustdar, S. & Srirama, S. N. Ccei-iot: Clustered and cohesive edge intelligence in internet of things. In *2022 IEEE International Conference on Edge Computing and Communications (EDGE)*, 33–40. https://doi.org/10.1109/EDGE55608.2022.00017 (2022).
17. Saleh, A. et al. Follow-me ai: Energy-efficient user interaction with smart environments. *IEEE Pervasive Comput.***24**, 32–42. https://doi.org/10.1109/MPRV.2025.3539421 (2025).
18. Yuan, J. et al. Grain crop yield prediction using machine learning based on UAV remote sensing: A systematic literature review. *Drones*. **8**, https://doi.org/10.3390/drones8100559 (2024).
19. Fang, Y., Kuang, Z., Wang, H., Lin, S. & Liu, A. Minimizing energy consumption of collaborative deployment and task offloading in two-tier UAV edge computing networks. *J. Syst. Archit.* 103511 (2025).
20. Wu, Y., Mu, X., Shi, H. & Hou, M. An object detection model AAPW-yolo for UAV remote sensing images based on adaptive convolution and reconstructed feature fusion. *Sci. Rep.***15**, 16214 (2025).
21. Ao, T. et al. Energy-efficient multi-uavs cooperative trajectory optimization for communication coverage: An madrl approach. *Remote Sens.***15**, https://doi.org/10.3390/rs15020429 (2023).
22. Westheider, J., Rückin, J. & Popović, M. Multi-UAV adaptive path planning using deep reinforcement learning. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 649–656. https://doi.org/10.1109/IROS55552.2023.10342516 (2023).
23. Fu, H. et al. A hierarchical path planning framework of plant protection uav based on the improved d3qn algorithm and remote sensing image. *Remote Sens.***17**, https://doi.org/10.3390/rs17152704 (2025).
24. Zhang, G. et al. Spiking neural networks in intelligent edge computing. *IEEE Consum. Electron. Mag.***14**, 66–75. https://doi.org/10.1109/MCE.2024.3506502 (2025).
25. Huang, C.-H., Chen, W.-T., Chang, Y.-C. & Wu, K.-T. An edge and trustworthy AI UAV system with self-adaptivity and hyperspectral imaging for air quality monitoring. *IEEE Internet Things J.***11**, 32572–32584 (2024).
26. Shan, J., Jiang, W., Huang, Y., Yuan, D. & Liu, Y. Unmanned aerial vehicle (UAV)-based pavement image stitching without occlusion, crack semantic segmentation, and quantification. *IEEE Trans. Intell. Transport. Syst.* (2024).
27. Yuan, Y. et al. Edge-cloud collaborative UAV object detection: Edge-embedded lightweight algorithm design and task offloading using fuzzy neural network. *IEEE Trans. Cloud Comput.***12**, 306–318 (2024).
28. Koubaa, A., Ammar, A., Abdelkader, M., Alhabashi, Y. & Ghouti, L. Aero: Ai-enabled remote sensing observation with onboard edge computing in uavs. *Remote Sens.***15**, https://doi.org/10.3390/rs15071873 (2023).
29. Han, Y., Duan, B., Guan, R., Yang, G. & Zhen, Z. Luffd-yolo: A lightweight model for UAV remote sensing forest fire detection based on attention mechanism and multi-level feature fusion. *Remote Sens.***16**, https://doi.org/10.3390/rs16122177 (2024).
30. Zhou, S., Zhou, H. & Qian, L. A multi-scale small object detection algorithm SMA-yolo for UAV remote sensing images. *Sci. Rep.***15**, 9255 (2025).
31. Yang, S. et al. Wheat yield prediction using machine learning method based on UAV remote sensing data. *Drones*. **8**, https://doi.org/10.3390/drones8070284 (2024).
32. Duan, Z., Liu, J., Ling, X., Zhang, J. & Liu, Z. Ernet: A rapid road crack detection method using low-altitude UAV remote sensing images. *Remote Sens.***16**, https://doi.org/10.3390/rs16101741 (2024).
33. Oubbati, O. S., Alotaibi, J., Alromithy, F., Atiquzzaman, M. & Altimania, M. R. A UAV-UGV cooperative system: Patrolling and energy management for urban monitoring. *IEEE Trans. Veh. Technol.***74**, 13521–13536. https://doi.org/10.1109/TVT.2025.3563971 (2025).
34. Alotaibi, J., Oubbati, O. S., Atiquzzaman, M., Alromithy, F. & Altimania, M. R. Optimizing disaster response with UAV-mounted RIS and hap-enabled edge computing in 6g networks. *J. Netw. Comput. Appl.* 104213 (2025).

35. Xu, H. et al. Edge computing resource allocation for unmanned aerial vehicle assisted mobile network with blockchain applications. *IEEE Trans. Wirel. Commun.***20**, 3107–3121 (2021).
36. Gao, X., Zhu, X. & Zhai, L. Aoi-sensitive data collection in multi-UAV-assisted wireless sensor networks. *IEEE Trans. Wirel. Commun.***22**, 5185–5197. https://doi.org/10.1109/TWC.2022.3232366 (2023).
37. Wan, P., Xu, G., Chen, J. & Zhou, Y. Deep reinforcement learning enabled multi-UAV scheduling for disaster data collection with time-varying value. *IEEE Trans. Intell. Transp. Syst.***25**, 6691–6702. https://doi.org/10.1109/TITS.2023.3345280 (2024).
38. Liao, Y. et al. Low-latency data computation of inland waterway USVS for RIS-assisted UAV MEC network. *IEEE Internet Things J.***11**, 26713–26726. https://doi.org/10.1109/JIOT.2024.3387017 (2024).
39. Wang, B. et al. Aav-assisted joint mobile edge computing and data collection via matching-enabled deep reinforcement learning. *IEEE Internet Things J.***12**, 19782–19800. https://doi.org/10.1109/JIOT.2025.3542025 (2025).
40. Raivi, A. M. & Moh, S. Jdaco: Joint data aggregation and computation offloading in UAV-enabled internet of things for post-disaster scenarios. *IEEE Internet Things J.***11**, 16529–16544. https://doi.org/10.1109/JIOT.2024.3354950 (2024).
41. Huang, J., Zhang, M., Wan, J., Chen, Y. & Zhang, N. Joint data caching and computation offloading in UAV-assisted internet of vehicles via federated deep reinforcement learning. *IEEE Trans. Veh. Technol.***73**, 17644–17656. https://doi.org/10.1109/TVT.2024.3429507 (2024).
42. Zhang, Y. et al. Joint trajectory and resource optimization for UAV and d2d-enabled heterogeneous edge computing networks. *IEEE Trans. Veh. Technol.***73**, 13816–13827. https://doi.org/10.1109/TVT.2024.3397335 (2024).
43. Wang, Z., Du, J., Jiang, C., Ren, Y. & Zhang, X.-P. UAV-assisted target tracking and computation offloading in USV-based MEC networks. *IEEE Trans. Mob. Comput.***23**, 11389–11405. https://doi.org/10.1109/TMC.2024.3396121 (2024).
44. Oubbati, O. S., Chaib, N., Lakas, A. & Bitam, S. On-demand routing for urban vanets using cooperating UAVS. In *2018 International Conference on Smart Communications in Network Technologies (SaCoNeT)*, 108–113 (IEEE, 2018).
45. Shao, R., Du, C., Chen, H. & Li, J. Sunet: Change detection for heterogeneous remote sensing images from satellite and UAV using a dual-channel fully convolution network. *Remote Sens.***13**, 3750 (2021).

Author contributions

Conceptualization, J.Z., Y.H., Y.T., M.S., L.W. and X.L.; methodology, J.Z., Y.H., L.W., M.S. and X.L.; data curation, J.Z., L.W. and X.L.; writing—original draft preparation, J.Z., Y.H., M.S., L.W. and X.L.; investigation, J.Z., Y.H., M.S., L.W. and X.L.; writing-review and editing, J.Z., L.W. and X.L.; visualization, J.Z., X.L., M.S. and L.W.; resources, Y.H., Y.T., L.W. and X.L.; supervision, Y.H., L.W. and X.L.; validation, J.Z., X.L., Y.H., Y.T. and L.W.; project administration, J.Z., M.S. and X.L.; visualization, J.Z., Y.T. and X.L.; All authors have read and agreed to the published version of the manuscript.

Funding

No funding was received for this study.

Declarations

Competing Interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Y.H. or X.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.