

Article

Fast and Efficient Data Collection Management Approach with Two-Layer UAV Network with Massive Sensor Nodes

Sanghyun Kim^{1,†} , Seunggho Yoo^{2,†} , Minjun Kim¹ , Ukhyun Jeong¹ , Wooyong Jung³ 
and Hwangnam Kim^{3,*} 

- ¹ Department of Data Engineering, Pukyong National University, Busan 48513, Republic of Korea; shkim1615@pukyong.ac.kr (S.K.); mjkim010305@pukyong.ac.kr (M.K.); 202655171@pukyong.ac.kr (U.J.)
² Department of Computer and Artificial Intelligence Engineering, Pukyong National University, Busan 48513, Republic of Korea; shyoo@pknu.ac.kr
³ School of Electrical Engineering, Korea University, Seoul 02841, Republic of Korea; jy17347@korea.ac.kr
* Correspondence: hnkim@korea.ac.kr
† These authors contributed equally to this work.

Abstract

Large-scale UAV data collection creates a tension among wide-area coverage, operational efficiency, and delivery continuity. Data must be continuously delivered to a base-station coordinator, but real-time replanning becomes increasingly difficult as the number of sensors and UAVs grows. Standard vehicle-routing methods slow down once routes have to be regenerated often, while reinforcement learning struggles with fixed-wing UAVs that cannot hover or turn sharply. We address this with a two-layer framework. In the lower layer, multirotor UAVs visit sensor nodes and buffer the collected payload until it is retrieved by a fixed-wing UAV. Their routes come from clustering the nodes and solving a capacitated vehicle routing problem within each cluster, with the cost biased toward older data and a short cooldown against immediate revisits. In the upper layer, fixed-wing UAVs deliver the buffered payload to the base-station coordinator, guided by a Multi-Agent Proximal Policy Optimization (MAPPO) policy that receives a local buffer-summary map and selected high-priority cells from a compact global summary. A spacing reward encourages separation before agents enter close-proximity states, instead of only penalizing collisions afterward. Component-level experiments show that the lower-layer planner handles up to 600 active routing targets within 1.3 s on average and that the age/cooldown objective improves freshness and revisit behavior. In integrated simulations with 1000 nodes, 32 multirotor UAVs, and 2 fixed-wing UAVs, the learned fixed-wing policy maintains collection performance comparable to a strong exclusive greedy baseline while recording no collision or persistent-proximity termination events over the reported data-generation-rate sweep. These results support the proposed framework as a scalable coordination-layer design for dynamic sensor workloads, where adaptive multirotor routing and motion-constrained fixed-wing retrieval are evaluated together under a shared data-generation workload.



Academic Editors: Alessandro Di Nuovo and Alessandro Lo Schiavo

Received: 12 May 2026

Revised: 18 June 2026

Accepted: 26 June 2026

Published: 3 July 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.

Keywords: Multi-UAV system; two-layer UAV network; data collection; cluster-based vehicle routing problem; multi-agent reinforcement learning; fixed-wing UAV; cooperative trajectory planning; UAV-assisted IoT

1. Introduction

Unmanned Aerial Vehicles (UAVs) are now widely used in various fields [1]. Once regarded primarily as hobbyist devices, UAVs have expanded into a wide range of practical applications. UAVs have made information-gathering tasks, such as search and rescue, 3D map generation, and site surveying, much more efficient. Most of these applications focus on collecting information in static environments, where conditions do not change frequently, and use a small number of UAVs. However, dynamic sensing environments change continuously, making it difficult for UAVs to gather information from such environments efficiently. In addition, to collect information over a wide area, it is crucial to deploy and manage a sufficient number of UAVs because UAV battery capacity and flight time are limited [2]. This limitation becomes even more critical in large-scale sensing scenarios, where timely and continuous data collection is essential.

Employing multiple UAVs to gather information in large-scale environments may provide an effective solution. However, several challenges must be addressed to utilize multiple UAVs over a large area. Above all, collisions among UAVs must be avoided. If UAVs collide with one another, they can no longer continue their mission, and additional UAVs may need to be deployed to complete the operation. To prevent such collisions, appropriate flight paths must be generated and assigned to individual UAVs. Most path-planning methods are derived from the Vehicle Routing Problem (VRP), which is NP-hard and requires excessive computation time as the number of UAVs increases. Because of this computational complexity, UAV flight paths cannot be updated frequently in large-scale environments. If the environment is stable and changes infrequently, this approach may be suitable. However, when the environment changes rapidly, the planned paths may no longer be appropriate. Moreover, when UAVs collect information in dynamic environments and deliver it to a base-station coordinator for environmental monitoring and analysis, the collected data must be transmitted promptly. This constraint requires frequent path updates to maintain data freshness and improve the quality of downstream environmental information.

To overcome these challenges, we propose a two-layer UAV coordination framework for efficient data collection in large dynamic environments and evaluate its coordination behavior through simulation. The system consists of two types of UAVs, multirotor UAVs and fixed-wing UAVs, each assigned to a different layer. In the lower layer, multirotor UAVs collect data from ground sensor nodes because of their high mobility and ability to change their routes with relatively little penalty. In the upper layer, fixed-wing UAVs collect data from the multirotor UAVs and relay the collected data to the base-station coordinator for environmental analysis and information generation. Although a homogeneous UAV fleet may simplify system design and operation, it is not well matched to the target environment considered in this study. Because fixed-wing UAVs must continuously maintain their flight speed to generate lift and cannot make sharp turns, they are not well suited for visiting ground nodes and collecting data along frequently changing flight paths. The multirotor UAVs can change their direction more easily than fixed-wing UAVs. However, their flight time is much shorter, resulting in a smaller coverage area. Therefore, we employ both types of UAVs together to compensate for the limitations of each platform.

A concrete setting that motivates this design is wide-area remote monitoring of a large region, on the order of 100 km², instrumented with many low-power ground sensors, where persistent long-range communication infrastructure is either unavailable or operationally undesirable, such as environmental monitoring, disaster-area assessment, and infrastructure inspection over terrain without reliable cellular or fixed-network coverage. In such settings, the commonly considered communication and sensing architectures each face a specific limitation. A low-power wide-area network or a multi-hop mesh requires every

static sensor to sustain a long-range or relay link toward the coordinator, which steadily drains the limited sensor energy budget as the monitored area and node count grow. A direct satellite or cellular uplink from each node removes the multi-hop burden but is constrained by infrastructure availability, per-node transmit power, and recurring cost. A homogeneous multirotor fleet can visit individual nodes but cannot cover a wide area because of its short flight time, whereas a homogeneous fixed-wing fleet offers endurance but cannot hover or make sharp turns and is therefore ill-suited to servicing individual ground nodes. The proposed two-layer design is intended for precisely this regime. Proximity-based multirotor collection removes the long-range-link requirement from the static sensors, while a small number of fixed-wing data mules exploit their endurance to retrieve the buffered payload over the wide area and deliver it to the coordinator. The second fixed-wing layer is therefore not an optional addition but the component that supplies wide-area retrieval that the energy-limited sensors and the short-endurance multirotors cannot provide on their own. The proposed architecture is not intended to replace mesh-network, low-power wide-area, direct-uplink, or relay-network deployments in general, but to provide a coordination-layer design for scenarios where sensing, buffering, and long-range retrieval are operationally separated.

In addition to the physical architecture of the proposed system, we also design an internal operational scheme suitable for large dynamic environments. To accelerate path generation for multirotor UAVs, we divide the target area into multiple subareas and generate paths separately, thereby significantly reducing the computational time required to solve the VRP and enabling continuous path updates in large dynamic environments. Because multirotor UAVs continuously move among the ground sensors, predetermined flight paths for fixed-wing UAVs cannot be derived effectively. Therefore, we devised a reinforcement-learning-based flight decision scheme and applied it to fixed-wing UAVs. Simulation results characterize the component-level design choices and the integrated collection-safety trade-off of the proposed two-layer coordination design against strong greedy baselines within the evaluated simulator. This paper targets the coordination layer of heterogeneous UAV data collection. Packet-level communication effects and high-fidelity autopilot dynamics are represented through coordination-level abstractions so that the study can isolate the coupled decision problem created by moving multirotor buffers and fixed-wing retrieval vehicles. Within this scope, the proposed framework contributes an integrated routing-retrieval design: a warm-started, age- and cooldown-aware clustered capacitated vehicle routing problem (CVRP) for local sensing, and a MAPPO-based fixed-wing retrieval policy that uses coordinator-derived buffer summaries, bounded waypoint actions, and spacing-aware coordination. The contributions of this work are as follows:

- We introduce a coordination-layer two-layer UAV framework that separates local sensing and buffering from long-range retrieval, allowing multirotor and fixed-wing UAVs to operate according to their platform-specific strengths.
- We develop a lower-layer replanning method that combines warm-started clustering, per-cluster CVRP solving, and age- and cooldown-aware routing costs to adapt routes as active sensing targets change over time.
- We design an upper-layer fixed-wing retrieval policy that turns compact coordinator-derived buffer summaries into bounded waypoint actions under fixed-wing motion and spacing constraints.
- We evaluate the coupled routing-retrieval behavior through component-level and integrated simulations, showing fast lower-layer replanning and competitive fixed-wing retrieval with no collision or persistent-proximity termination events in the reported operating range.

2. Background

This section describes the multi-UAV data collection problem considered in this study.

The use of multiple UAVs for data collection has been widely studied in various application domains, including IoT-based sensor networks, environmental monitoring, and disaster response [3–5]. This problem goes beyond simply visiting predefined locations; rather, it involves exploring meaningful information and collecting it efficiently while accounting for changing environmental conditions.

In dynamic sensing environments, the spatial distribution and importance of data may vary over time. Accordingly, UAVs must determine exploration priorities and collection order under limited flight time and resources, and a balance between exploration efficiency and collection performance becomes essential [6].

In particular, unlike the single-UAV setting, multi-UAV environments require the simultaneous consideration of inter-agent cooperation, role allocation, and collision avoidance, which further increases problem complexity [7]. As a result, the multi-UAV data collection problem extends beyond simple path planning to include sequential decision-making and cooperative operation in dynamic environments. Owing to these characteristics, both path-optimization-based [8–10] and learning-based [11] approaches have been extensively studied, and each offers distinct advantages depending on the problem setting.

3. Related Work

In this section, we review existing literature on multi-UAV coordination, hierarchical systems, and heterogeneous systems.

3.1. Optimization- and Learning-Based Approaches for Multi-UAV Coordination

Multi-UAV exploration and data collection problems have traditionally been formulated as optimization problems such as the Vehicle Routing Problem (VRP), Traveling Salesman Problem (TSP), and Multi-Agent Path Finding (MAPF) [11–13]. These approaches enable the systematic computation of UAV trajectories to reduce total travel distance or operation time, and they can also be effective in terms of computational efficiency when combined with clustering or heuristic methods. In particular, when the target environment is relatively static or can be divided into regional subproblems, such methods can provide effective baselines.

Drone-assisted routing studies further show that vehicle-routing variants can represent cooperation between ground vehicles, depots, and UAVs, including delivery or service problems in which UAVs are inserted into a larger route structure [14,15]. These cited formulations are useful for designing optimized schedules, but they assume that the entities to be served are known at planning time or that the routing problem can be solved as a relatively self-contained optimization instance. In large dynamic sensing scenarios, by contrast, the amount and age of pending data, the positions of collection vehicles, and the retrieval opportunities available to fixed-wing UAVs change continuously.

When the environment changes continuously, precomputed paths may need to be updated repeatedly, which can increase computational burden and create difficulties for real-time operation. To ease this burden, incremental and partial-replanning strategies have been explored, including incremental graph-search methods that reuse prior computation and evolutionary rolling-horizon schemes that re-optimize short path segments rather than entire trajectories [16–18].

In this context, reinforcement learning-based approaches have attracted increasing attention because they can respond more flexibly to changing situations by learning policies through interaction with the environment. In practice, algorithms such as PPO, MADDPG, and MAPPO have been applied in various studies on multi-agent and multi-UAV cooperation [19–23]. Nevertheless, reinforcement-learning-based approaches also require careful design in terms of training stability, reward design, and sample efficiency. This issue becomes even more important in systems with strong physical constraints, where learning valid actions in a stable manner is particularly challenging [24].

3.2. Hierarchical and Heterogeneous Multi-UAV Systems

In large-scale Unmanned Aerial Vehicle (UAV) systems, hierarchical architectures have frequently been adopted to improve communication efficiency and scalability [25,26]. For instance, previous studies have proposed structures where UAVs are assigned distinct roles, utilizing specific agents to collect data from others and deliver it to a central coordinator [27,28]. While these hierarchical and non-terrestrial-network designs are highly effective in reducing communication burdens, they primarily focus on network control, spectrum management, or communication objectives rather than the joint routing-and-retrieval behavior of heterogeneous UAV platforms.

To maximize data collection efficiency, recent literature has increasingly explored heterogeneous multi-agent systems. However, the configuration of such teams has been predominantly limited to cooperation between UAVs and Unmanned Ground Vehicles (UGVs) [29,30], or cooperation between UAVs with flight range constraints and ground vehicles [31]. Even in literature focusing strictly on heterogeneous UAV-to-UAV systems, heterogeneity typically stems from simple variations in battery capacity, computing power, or speed, resulting only in straightforward task allocation [32,33]. Instances where UAVs with fundamentally distinct maneuvering characteristics and aerodynamic constraints are organically combined remain scarce.

Consequently, much of the existing UAV literature relies on homogeneous platforms, particularly quadrotor-based systems. Although homogeneity offers advantages in design consistency and control simplicity, it limits the extent to which role allocation can exploit platform-specific advantages. For example, homogeneous rotary-wing environments frequently suffer from chronic battery capacity limitations. Prior studies have attempted to circumvent this constraint through ad-hoc methods, such as imposing strict mission time limits or optimizing local trajectories [34]; however, these approaches act as a performance bottleneck in large-scale scenarios since they do not directly overcome the low energy capacity itself. Transitioning to a heterogeneous environment by introducing fixed-wing UAVs—which offer superior long-endurance flight and wide-area coverage [35,36]—provides a viable alternative to mitigate this critical energy issue.

Nevertheless, operating aerial vehicles with highly distinct dynamic characteristics within a single operational layer introduces significant control and decision-making challenges [37]. Fixed-wing UAVs exhibit strict physical constraints, such as minimum flight speed, limited turning radii, and the inability to hover or move backward. When these platforms are controlled directly, particularly in reinforcement learning (RL) approaches, the action space is heavily restricted by these dynamics, making stable exploration difficult [38]. Furthermore, issues like sparse rewards or convergence to undesirable strategies (e.g., reward hacking) are exacerbated in complex multi-agent environments, severely degrading overall training stability [39,40]. Operating them flatly in a single layer causes the physical limitations of each platform to become complexly entangled.

To alleviate these challenges, introducing a hierarchical architecture into the heterogeneous multi-UAV environment can be useful. As highlighted in [41], a hierarchical approach mitigates learning difficulties, enables the effective use of the complementary advantages inherent in heterogeneous systems, and provides a structural basis for multi-agent coordination. The present work leverages this paradigm, positioning the hierarchy as a coordination mechanism in which platform heterogeneity is treated as an operational structure rather than an implementation detail. Within this structure, the upper layer uses coordinator-derived buffer summaries and compact priority cues to select retrieval-oriented waypoint actions, with the goal not of computing globally optimal trajectories but of preserving high retrieval performance under fixed-wing motion and spacing constraints. This avoids short-sighted approaches that determine immediate local paths based solely on current buffer volumes.

The proposed framework targets a distinct routing-retrieval interface. The hierarchy limits the multirotor layer to local sensing, buffering, and lightweight metadata exchange, while assigning high-mobility fixed-wing data mules to payload retrieval. Our approach combines the computational efficiency of optimization with the adaptive flexibility of learning by partitioning the overall task into a two-layer framework, which consists of clustering and CVRP routing for multirotors in the lower layer and a MAPPO policy for fixed-wing data retrieval in the upper layer. This positioning distinguishes our work from studies that optimize only routing, only relay communication, or only learned motion control.

We emphasize that the individual building blocks of this framework, namely hierarchical UAV architectures, heterogeneous teams, clustered routing, and reinforcement-learning-based coordination, are established in the literature, and we do not claim any of them as novel in isolation. The contribution is the coupled decision problem that arises when these elements operate together under platform-specific motion constraints, which prior work has largely avoided by treating only one sub-problem at a time. Three aspects distinguish the present design. First, the routing-retrieval interface between moving multirotor buffers and fixed-wing retrieval vehicles is itself the object of study, because the lower layer produces a time-varying buffer field that directly determines the observation and reward of the upper-layer policy, so the two layers are evaluated as a coupled system rather than in isolation. Second, the heterogeneity addressed here is not a quantitative variation in battery, speed, or computing power that reduces to straightforward task allocation, as in much prior heterogeneous-UAV work, but a combination of platforms with fundamentally different maneuvering characteristics, including the inability of fixed-wing UAVs to hover and their minimum turning radius. The bounded waypoint-to-motion conversion and the spacing-aware reward are designed specifically to make learned coordination feasible under these constraints. Third, the lower-layer replanning combines warm-started clustering with a separated age- and cooldown-aware cost so that routes adapt to time-varying demand without sacrificing short-term stability, rather than applying a generic clustering-plus-CVRP pipeline. The component-level study in Section 6.1 shows that these design choices are load-bearing rather than incidental. A single high-priority target is not sufficient and leads to repeated termination events, and the spacing-aware reward needs an appropriate separation distance to keep the fixed-wing pair clear of close-proximity states. These results indicate that the specific coupling of these elements, rather than the mere presence of the four building blocks, is what produces termination-free high-throughput retrieval.

4. Proposed Approach

In this section, we present the overall structure of the proposed two-layer UAV framework and describe its components.

4.1. Overview of Two-Layer UAV Framework

We designed the proposed two-layer UAV framework to gather information from large dynamic environments. The overview of our proposed framework is presented as Figure 1. The framework consists of two layers: the multirotor layer and the fixed-wing layer. The lower multirotor layer collects data from ground nodes that gather data from the surrounding environment. The multirotor UAVs visit nearby ground nodes to collect accumulated data. The upper fixed-wing layer delivers the data collected by the multirotor UAVs to the base-station coordinator. Because the positions of the multirotor UAVs and their buffered data continuously change, conventional path-generation algorithms, including the traveling salesman problem (TSP) and its variants, are not suitable as a complete upper-layer retrieval strategy for the fixed-wing UAVs. To schedule the retrieval behavior of the fixed-wing UAVs, we employ an RL-based planner that outputs bounded waypoint commands based on local data and compact priority information. The detailed operational procedure and structure of each layer and component are described in the following subsections.

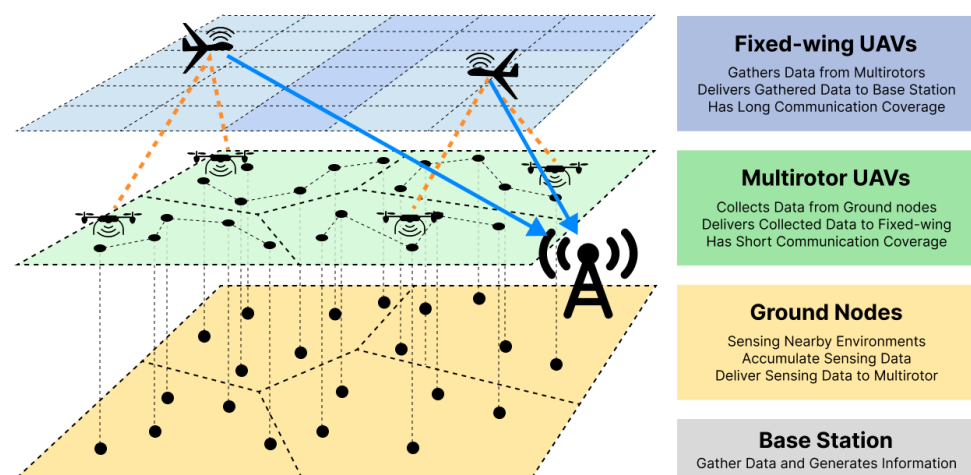


Figure 1. Overview of the proposed two-layer UAV framework.

4.2. Design Goal

The proposed two-layer framework is designed with three operational objectives in mind, which together motivate the structural choices described in the remainder of this section.

The first objective is scalability with respect to the number of ground nodes. As the monitored area grows and the number of nodes increases, any scheme that attempts to maintain a continuous end-to-end connection between every node and a base-station coordinator quickly becomes infeasible in terms of communication bandwidth and energy consumption. The framework therefore seeks to decouple sensing from delivery, so that adding more nodes does not directly translate into a proportional increase in network-wide signaling.

The second objective is responsiveness in dynamic environments. Because the spatial distribution of pending data changes continuously as new measurements are generated and previously collected data is removed, any precomputed plan becomes outdated within a short time window. The framework must therefore support periodic replanning at both layers, with per-cycle computation kept light enough to remain tractable in large-scale settings.

The third objective is the physical realizability of the planned behavior. Multirotor and fixed-wing UAVs differ substantially in their maneuvering characteristics, and a single control paradigm is unlikely to exploit both platforms efficiently. Accordingly, the framework assigns each platform to the role for which it is better suited, rather than forcing them into a homogeneous operating pattern.

These three objectives, scalability, responsiveness, and platform-aware role allocation, jointly motivate the two-layer structure. In the lower layer, a fleet of multirotor UAVs visits ground nodes and buffers the collected data locally, so that the node-to-UAV link is activated only when a vehicle is in proximity. In the upper layer, a small number of fixed-wing UAVs traverse the operational area, retrieve buffered data from the multirotor UAVs through contact-range pickup events, and relay it to the base-station coordinator. Separating the roles in this way reduces the burden of maintaining long-range links on the multirotor UAVs and allows the fixed-wing UAVs to exploit their endurance for wide-area coverage.

The framework uses a metadata-assisted data-mule abstraction. Sensed payloads remain in multirotor buffers, while the coordinator receives lightweight status metadata, including position, route status, buffer amount, oldest-buffer age, and pickup requests. Fixed-wing UAVs also communicate with the coordinator through the same low-rate telemetry/control channel by reporting their position, heading, and timestamp. The coordinator converts reported multirotor buffer metadata into a buffer-summary map, a fleet-state summary, and compact retrieval cues used by the fixed-wing policy. Inter-agent state information in the policy observation is therefore coordinator-relayed telemetry metadata, not direct air-to-air communication or payload exchange.

In the present simulator, payload exchange is implemented as a contact-pickup event: when a fixed-wing UAV enters the contact range of a buffered multirotor UAV, the simulator records the pickup and updates the multirotor buffer state. When the fixed-wing UAV reaches the central delivery region, the retrieved payload is counted as delivered at the base-station coordinator. This implementation keeps the upper-layer retrieval field tied to moving multirotor buffers, while packet-level transfer dynamics remain deployment-oriented extensions of the same pickup interface.

The remainder of this section describes how each layer realizes its role. The lower-layer procedure is presented first, with an emphasis on how the exploration and collection tasks are distributed among multirotor UAVs and how each vehicle's path is organized in response to continuously changing demand. The fixed-wing layer, which takes the state of the lower layer as its operating environment, is then described in the following section.

4.3. Multirotor UAVs: Data Collection

This section describes the data collection operations of the lower-layer multirotor UAVs. In a large-scale exploration environment, the number of targets, the number of active vehicles, and field conditions change continuously, and a systematic procedure together with a fast decision-making structure is required to maintain mission stability. The subsection is organized into two parts. The operational procedure is described first from a field-operation perspective, followed by the algorithmic design of task allocation and path planning that supports that procedure.

4.3.1. Operational Procedure

Multirotor operations proceed through a cyclic structure consisting of mission initialization, zone assignment, route-based exploration execution, status monitoring, and re-assignment or replanning. Each vehicle collects data by traversing waypoints within its

assigned zone, and the collected data is held in the local buffer until it is retrieved by a fixed-wing UAV of the upper layer.

During operations, mission progress and pending-data distribution are periodically summarized for replanning. The scheduler updates zone assignments and routes when the task distribution changes, so that exploration gaps can be reduced within the coordination-level model. The interval between replanning events is controlled by a single configuration parameter, so that the balance between adaptivity and route stability can be tuned according to how rapidly the workload evolves. The experiments use this cyclic replanning model to evaluate how lower-layer routes adapt as pending data changes over time.

4.3.2. Task Allocation and Path Planning

The lower-layer routing pipeline is organized into two sequential stages, task allocation and per-vehicle path planning. Given a large number of ground nodes and a fleet of multirotor UAVs, the first stage partitions the deployed ground nodes into per-vehicle service regions, and the second stage routes only the pending-task demand inside each assigned region under route-time and capacity constraints. The overall workflow is periodically re-executed so that the plan remains consistent with the current distribution of pending data.

Task Allocation via Warm-Started Clustering

Let $\mathcal{N} = \{n_1, n_2, \dots, n_{|\mathcal{N}|}\}$ denote the set of deployed ground nodes, where each node n_i has a fixed position $p_i \in \mathbb{R}^2$. At replanning time t , the subset of nodes that have pending data is

$$\mathcal{N}_t^{\text{act}} = \{n_i \in \mathcal{N} \mid \mathcal{T}_i(t) \neq \emptyset\}.$$

Let $|\mathcal{M}|$ denote the number of active multirotor UAVs. The clustering stage partitions the full deployed node set $\mathcal{N}$ into $|\mathcal{M}|$ disjoint subsets $\{\mathcal{C}_1, \dots, \mathcal{C}_{|\mathcal{M}|}\}$ that minimize the intra-cluster dispersion

$$\min_{\{\mathcal{C}_k\}, \{\mu_k\}} \sum_{k=1}^{|\mathcal{M}|} \sum_{n_i \in \mathcal{C}_k} \|p_i - \mu_k\|_2^2,$$

where μ_k denotes the centroid of cluster k . Each cluster is then associated with a single multirotor UAV. The subsequent CVRP stage ignores nodes without pending tasks, so only $\mathcal{C}_k \cap \mathcal{N}_t^{\text{act}}$ contributes nonzero demand to vehicle k at that replanning cycle.

Because the set of nodes with pending data evolves over time as new tasks are generated and previously collected tasks are removed, assignment and routing are recomputed at fixed replanning intervals. Running the clustering from scratch at every cycle, however, tends to produce abrupt changes in cluster boundaries and causes multirotor UAVs to repeatedly switch between service regions. To mitigate this, a warm-start scheme is used in which the centroids obtained in the previous replanning cycle are reused as the initial centers for the current cycle, and Lloyd's algorithm is executed for a fixed number of iterations. Only at the very first cycle, or when the number of active vehicles changes, are the centers re-initialized from deployed nodes. This design keeps cluster identities largely consistent across cycles while still allowing the per-vehicle routed demand to adapt to changes in the spatial distribution of pending tasks.

Per-Vehicle Path Planning as a CVRP

For each multirotor UAV, the visiting order of the active nodes in its assigned cluster is planned as a CVRP. For a given vehicle k , let $\mathcal{C}_k^{\text{act}} = \mathcal{C}_k \cap \mathcal{N}_t^{\text{act}}$ be the active routed-demand set and let $\mathcal{V}_k = \mathcal{C}_k^{\text{act}} \cup \{0, d\}$ denote the set of locations, where 0 is the route start used for the current planning call and d is the operational rendezvous or turnaround point used for

route reset, battery-related return, and coordination with the retrieval layer. The location d belongs to the coordination-layer route abstraction and provides a reset or rendezvous reference for the multirotor plan. Tasks generated at the same node are aggregated into a single demand

$$q_i = \sum_{\tau \in \mathcal{T}_i} s_\tau, \quad i \in \mathcal{C}_k^{\text{act}},$$

where $\mathcal{T}_i$ is the set of pending tasks at node n_i and s_τ denotes the size of task τ in bytes. Let Q be the per-vehicle buffer capacity, $\mathcal{R}_k$ the set of duplicated depot-start routes allowed for vehicle k , and $\mathcal{A}_k^{(r)}$ the candidate arc set for route $r \in \mathcal{R}_k$. Let $T_{\max}^{(k)}$ be the remaining route-time budget at the time of planning. The following equations summarize the objective and nonstandard constraints used in our OR-Tools routing implementation. Standard routing feasibility constraints, including depot continuity, route connectivity, flow conservation, and the visit-arc linking constraints that connect $y_i^{(r)}$ to the selected arcs $x_{ij}^{(r)}$, are enforced by the routing solver. The CVRP for vehicle k can then be written as

$$\min \sum_{r \in \mathcal{R}_k} \sum_{(i,j) \in \mathcal{A}_k^{(r)}} c_{ij} x_{ij}^{(r)} \quad (1)$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{C}_k^{\text{act}}} q_i y_i^{(r)} \leq Q, \quad \forall r \in \mathcal{R}_k, \quad (2)$$

$$\sum_{(i,j) \in \mathcal{A}_k^{(r)}} t_{ij} x_{ij}^{(r)} \leq T_{\max}^{(k)}, \quad \forall r \in \mathcal{R}_k, \quad (3)$$

$$\sum_{r \in \mathcal{R}_k} y_i^{(r)} \leq 1, \quad \forall i \in \mathcal{C}_k^{\text{act}}, \quad (4)$$

where $x_{ij}^{(r)}$ is a binary variable indicating whether arc (i, j) is traversed during duplicated route r , $y_i^{(r)}$ indicates whether node i is served during route r , and t_{ij} is the travel time on arc (i, j) . Constraints (2) and (3) encode the per-route capacity and route-time limits, while constraint (4) ensures that each node is served at most once across the entire plan. The duplicated-route representation provides a coordination-layer abstraction for mid-plan returns to the rendezvous point; cumulative battery recharge or replacement can be layered onto this rendezvous interface in deployment-oriented extensions.

A distinctive feature of the proposed formulation lies in the arc cost c_{ij} , which departs from a purely distance-based metric in two respects. First, nodes holding older data are prioritized through an age-weighted term, because their contents contribute less to system-level freshness the longer they are delayed. For an arc terminating at node j , let $a_j = t - \min_{\tau \in \mathcal{T}_j} g_\tau$ be the age of the oldest pending task at j , where g_τ is the generation time of task τ and t is the current time. Second, a cooldown penalty is applied to suppress short-interval revisits to recently serviced nodes, which otherwise occur frequently when residual data is negligible. Combining these two effects, the arc cost is defined as

$$c_{ij} = \frac{d_{ij}}{1 + \alpha a_j} + \rho [t - v_j < \Delta_{\text{cool}}],$$

where d_{ij} is the Euclidean distance between locations i and j , v_j is the last visit time of node j , Δ_{cool} is the revisit cooldown window, and ρ and α are tuning coefficients controlling the penalty magnitude and the age weighting, respectively. The coefficient α has inverse-time units, while ρ is expressed in distance-equivalent cost units so that the two terms can be added. When node j is the depot or a non-task route-start location, both the penalty and the age term vanish. This cost structure favors nodes that are simultaneously close to the current route and carry aged data, while discouraging routes that return to a node shortly

after it has been visited. The distance term is divided by an age-dependent weight so that nodes with older pending data become more attractive even when they are not the nearest candidates. The cooldown penalty is then added separately, rather than being divided by the same age weight, to keep recent-revisit suppression active even for aged nodes. This separation makes the first term a freshness-aware distance weight and the second term an independent anti-revisit mechanism.

To accommodate cases in which the total demand of a cluster exceeds the vehicle capacity, the formulation permits multiple duplicated routes per vehicle, so that the same multirotor UAV can return to the rendezvous point mid-plan and resume service on subsequent nodes in the route abstraction. In addition, a disjunction with a sufficiently large drop penalty is associated with every node, which allows the solver to skip a node when no feasible assignment exists under the capacity and time constraints, rather than returning an infeasible problem. The CVRP formulation keeps the capacity interface explicit, although in the reported workload the route-time bound and active-demand distribution are the dominant practical constraints. The resulting CVRP is solved with the OR-Tools Routing Library [42], and the solution is converted into a sequence of waypoints that includes service stops at nodes and turnaround or rendezvous stops.

Periodic Replanning and Continuity

The two-stage allocation-and-routing procedure is repeated every T_{rp} seconds. At each replanning instant, the current position of each multirotor UAV, its remaining flight time, the up-to-date list of pending tasks, and the most recent node visit times are passed to the planner, and the warm-started clustering and CVRP are solved in order. To avoid abrupt direction changes when a multirotor UAV is already en route, the waypoint that the vehicle is currently traveling toward is preserved as the first element of the new plan, and the remaining route is recomputed starting from that waypoint. This ensures that replanning remains compatible with the multirotor UAV's current heading while continuously incorporating newly generated tasks and updated cluster assignments. As the cluster partition, the age-weighted arc cost, and the cooldown penalty are all recomputed at every cycle, the lower-layer routes adapt to the temporal evolution of pending data without sacrificing short-term operational stability.

Algorithm 1 summarizes the entire lower-layer planning procedure described above. The algorithm is invoked once per replanning cycle and jointly produces, for every active multirotor UAV, an updated waypoint sequence that reflects the current distribution of pending data, the most recent visit history, and the remaining flight-time budget.

This coordinator-mediated exchange defines the interface between the two layers. During the lower-layer cycle, multirotor UAVs execute their current routes and report telemetry metadata, including their positions and buffered-data summaries, to the base-station coordinator. The lower layer produces a time-varying buffer field through multirotor sensing and routing, while the upper layer receives compact retrieval cues derived from that field and selects fixed-wing waypoint actions. Fixed-wing UAVs also report their own fleet-state telemetry back to the coordinator, which relays compact relative-state summaries used by the policy. When a fixed-wing UAV reaches the communication-radius contact condition of a buffered multirotor UAV, the model records a payload pickup event and updates the buffered-data state. The integrated experiment evaluates this coupled interface under a shared data-generation workload.

Algorithm 1: Lower-Layer Replanning Cycle for Multirotor Routing

Input: Node set $\mathcal{N}$ with pending tasks $\{\mathcal{T}_i\}$; multirotor set $\mathcal{M}$; previous centroids $\{\mu_k^{\text{prev}}\}$; current time t ; node visit history $\{v_j\}$; replanning period T_{rp}

Output: Route $R_k = (w_{k,1}, w_{k,2}, \dots)$ for each multirotor UAV $k \in \mathcal{M}$

// Stage 1: Warm-started clustering of deployed nodes

```

1  $\mathcal{N}^{\text{act}} \leftarrow \{n_i \in \mathcal{N} \mid \mathcal{T}_i \neq \emptyset\}$ 
2 if  $\mathcal{N}^{\text{act}} = \emptyset$  then
3   | assign empty routes to all vehicles and return
4 end
5 if  $\{\mu_k^{\text{prev}}\}$  is available and  $|\mathcal{M}|$  unchanged then
6   | initialize centers  $\{\mu_k^{(0)}\} \leftarrow \{\mu_k^{\text{prev}}\}$ 
7 else
8   | initialize centers from deployed nodes  $\mathcal{N}$ 
9 end
10 for  $\ell = 1$  to  $L$  do
11   | assign each  $n_i \in \mathcal{N}$  to cluster  $\arg \min_k \|p_i - \mu_k^{(\ell-1)}\|_2$ 
12   | recompute centroids  $\{\mu_k^{(\ell)}\}$  from current assignments
13 end
14  $\{\mathcal{C}_k\} \leftarrow$  final cluster partition; store  $\{\mu_k^{(L)}\}$  for the next cycle

// Stage 2: Per-vehicle CVRP
15 foreach multirotor UAV  $k \in \mathcal{M}$  do
16   | if  $k$  is en route to waypoint  $w_k^*$  then
17     |  $s_k \leftarrow w_k^*$  // preserve the in-flight waypoint
18   | else
19     |  $s_k \leftarrow$  current position of  $k$ 
20   | end
21    $\mathcal{C}_k^{\text{act}} \leftarrow \mathcal{C}_k \cap \mathcal{N}^{\text{act}}$ 
22    $\mathcal{V}_k \leftarrow \mathcal{C}_k^{\text{act}} \cup \{s_k, \text{depot } d\}$ 
23   foreach node  $i \in \mathcal{C}_k^{\text{act}}$  do
24     |  $q_i \leftarrow \sum_{\tau \in \mathcal{T}_i} s_\tau$ 
25     |  $a_i \leftarrow t - \min_{\tau \in \mathcal{T}_i} g_\tau$ 
26   | end
27   foreach arc  $(i, j) \in \mathcal{A}_k$  do
28     | if  $j$  is depot or route-start location then
29       |  $c_{ij} \leftarrow d_{ij}$ 
30     | else
31       |  $c_{ij} \leftarrow \frac{d_{ij}}{1 + \alpha a_j} + \rho [t - v_j < \Delta_{\text{cool}}]$ 
32     | end
33   | end
34    $T_{\text{max}}^{(k)} \leftarrow$  remaining flight-time budget of  $k$ 
35   solve CVRP over  $\mathcal{V}_k$  with cost  $\{c_{ij}\}$ , demand  $\{q_i\}$ , capacity  $Q$ , time bound  $T_{\text{max}}^{(k)}$ , and per-node drop disjunction
36   convert the CVRP solution into waypoint sequence  $R_k^{\text{new}}$ 
37    $R_k \leftarrow R_k^{\text{new}}$ 
38 end
39 return  $\{R_k\}_{k \in \mathcal{M}}$ 

```

4.4. Fixed-Wing UAVs: Data Retrieval and Delivery

This section describes the data-retrieval operations of the upper-layer fixed-wing UAVs. The fixed-wing layer treats the buffered data distribution produced by the lower-layer multirotors as its retrieval field. Because the spatial pattern of buffered data shifts continuously as multirotors move, and because fixed-wing platforms cannot hover or make sharp turns, a different decision-making structure from the lower-layer is required, one that reacts to a changing target field while respecting the dynamic constraints of the platform. The subsection is organized into two parts. The operational procedure is described first from a field-operation perspective, followed by the learning-based design of observation, action, and reward that supports that procedure.

4.4.1. Problem Formulation

The fixed-wing layer treats the buffered data distribution of the multirotor layer as its data-retrieval field. Because directly using the full coordinator-side buffer-summary map would increase the observation dimension, the metadata-derived retrieval information is compressed into a 10×10 grid-based map. For the multirotor set $\mathcal{M}$, each cell (u, v) stores the reported buffered-data amount

$$\hat{D}_t(u, v) = \sum_{m \in \mathcal{M}} \hat{b}_m^t [p_m^t \in \text{cell}(u, v)],$$

where $\hat{b}_m^t$ denotes the buffered-data amount reported by multirotor m through telemetry metadata. The map $\hat{D}_t$ is therefore a coordinator-side summary of where buffered payload is located, not the payload itself. The objective of each fixed-wing UAV is to reduce the remaining buffered data within a limited flight time while maintaining safe inter-agent spacing.

The state of each agent i is defined as follows:

$$x_i^t = [p_i^t, \theta_i^t, \omega_i^t],$$

where $p_i^t \in \mathbb{R}^2$ denotes position, θ_i^t denotes heading angle, and ω_i^t denotes angular velocity.

In particular, fixed-wing UAVs have a minimum turning radius and continuous dynamic constraints, making cooperative exploration and control more difficult than for multirotor UAVs. Furthermore, in a multi-agent environment, agents may concentrate on the same high-value regions, which can lead to reduced exploration efficiency and increased collision risk. Therefore, this study formulates a reinforcement learning problem that simultaneously considers data collection efficiency and inter-agent spacing maintenance.

4.4.2. Overall Architecture

In this study, a hierarchical design that decouples the observation structure from the action structure is used to support efficient learning of fixed-wing waypoint decisions under bounded forward-flight dynamics. The proposed architecture consists of four components, and the overall training framework is illustrated in Figure 2. The observation combines a local buffer-summary map with coordinator-provided priority targets. The decision-making policy network is trained with MAPPO. The action output is a waypoint-shaped command, which is interpreted by the simulator through a bounded waypoint-to-motion conversion layer.

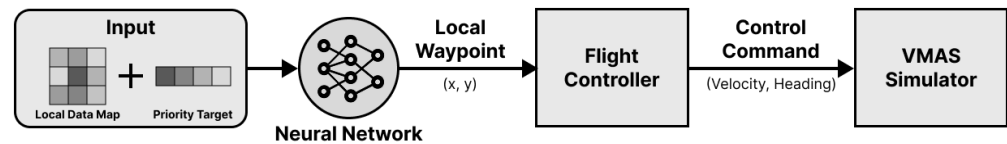


Figure 2. Training framework for fixed-wing UAV learning with a waypoint-to-motion interface.

MAPPO is used as the training method for the policy network responsible for decision-making. MAPPO effectively addresses high-dimensional problems through the powerful representation learning of deep learning [43,44], and adopts the Actor-Critic approach to mitigate the high-variance problem of policy-based learning and support more stable policy optimization [45–47]. In particular, by employing a CTDE (Centralized Training, Decentralized Execution) structure that utilizes full state information during training while using only individual observations during execution, the persistent non-stationarity problem of multi-agent environments is alleviated [48]. Owing to these characteristics, the effectiveness of MAPPO in multi-agent environments has been demonstrated in prior studies [21], and it has been widely adopted in many MARL frameworks [22,23,49,50].

The key to generating the movement of fixed-wing UAVs is the decoupling of high-level waypoint choice from the simulator’s motion update. Reinforcement learning serves as the high-level decision-making model that determines a relative waypoint command, while the bounded conversion layer maps this command to the next fixed-wing motion state. This places the contribution at the coordination-policy level: the learned policy selects retrieval-oriented waypoint commands, while the bounded conversion layer enforces the fixed-wing motion envelope used consistently across all evaluated methods.

The proposed observation structure has two roles: a local buffer-summary map responsible for close-range control and immediate response, and priority targets responsible for guiding global retrieval direction. This observation structure does not use the full buffer-summary map as direct input, but instead selectively provides only the most important information. In other words, coordinator-side retrieval information is retained while effectively reducing the observation space. This encourages the policy to perform decision-making at different levels without requiring an explicit hierarchical reinforcement learning structure.

4.4.3. Action Space and Dynamics Structure

Waypoint-Based Action Definition

In conventional end-to-end reinforcement learning, agents directly output low-level control inputs such as velocity or steering angle. However, for fixed-wing UAVs, this approach leads to a complex action-outcome relationship due to turning radius and directional constraints, and learning becomes unstable owing to continuous dynamics. This significantly degrades sample efficiency.

The policy is therefore defined to output a waypoint-shaped action:

$$a_i^t \in [-1, 1]^2.$$

This is interpreted as the target movement direction (waypoint) relative to the current position.

Bounded Waypoint-to-Motion Conversion

The fixed-wing layer is implemented in a modified VMAS simulator. The policy does not directly output low-level actuator, velocity, or steering commands. Instead, it outputs a normalized relative waypoint. The fixed-wing dynamics motion module of

the modified VMAS converts this waypoint into a bounded forward-flight update by maintaining positive forward speed, limiting the heading response, and disallowing hover, reverse motion, or instantaneous point interception. Therefore, a waypoint located behind the vehicle or inside the implied turning range only changes the desired bearing; the realized vehicle motion remains a bounded forward-flight turn. The same conversion settings are used for all learned-policy evaluations.

At the coordination level, the bounded conversion can be summarized by the fixed-wing module's motion update. Given the normalized waypoint action a_i^t , the simulator first constructs a relative waypoint

$$w_i^t = p_i^t + L_w a_i^t,$$

where L_w is the waypoint scaling length. The desired bearing is

$$\psi_i^t = \text{atan2}(w_{i,y}^t - p_{i,y}^t, w_{i,x}^t - p_{i,x}^t).$$

The motion-conversion module converts the bearing error into a steering command δ_i^t and clips it by the steering-command limit:

$$\delta_i^t = \text{clip}(\delta_{\text{cmd}}(\psi_i^t - \theta_i^t), -\delta_{\text{max}}, \delta_{\text{max}}).$$

The simulator then applies the fixed-wing module's motion update using the front and rear length parameters l_f and l_r , the forward speed v_i^t , and the simulation time step Δt :

$$\beta_i^t = \text{atan2}\left(\tan(\delta_i^t) \frac{l_r}{l_f + l_r}, 1\right),$$

$$\dot{\theta}_i^t = \frac{v_i^t}{l_f + l_r} \cos(\beta_i^t) \tan(\delta_i^t),$$

$$\theta_i^{t+1} = \theta_i^t + \dot{\theta}_i^t \Delta t,$$

$$p_i^{t+1} = p_i^t + v_i^t \Delta t \begin{bmatrix} \cos(\theta_i^{t+1} + \beta_i^t) \\ \sin(\theta_i^{t+1} + \beta_i^t) \end{bmatrix}.$$

Under the reported fixed-layer configuration, $l_f = l_r = 0.025$ simulator units and $\delta_{\text{max}} = 20^\circ$, which gives

$$R_{\text{min}} = \frac{l_f + l_r}{\cos(\beta_{\text{max}}) \tan(\delta_{\text{max}})} \approx 0.140 \text{ simulator units} \approx 140 \text{ m},$$

where β_{max} denotes the maximum value of β_i^t under the steering-command limit δ_{max} , and the final conversion uses the fixed-layer scale of 1000 m per simulator unit. The reported speed range of 20–40 m/s determines the forward displacement per step, whereas the approximate 140 m value follows from the steering clamp and simulator length scale. Therefore, a waypoint located behind the aircraft or closer than the implied turning radius is not treated as a hover, reverse, or instantaneous point-interception command. It only changes the desired bearing, and the realized motion remains bounded by the forward-speed and steering-response limits.

The 4 m collision termination distance is a simulator-level center-to-center overlap criterion: each fixed-wing UAV is represented by a circular collision envelope with a 2 m radius, so two vehicles are counted as colliding when their centers are closer than 4 m. The fixed-wing motion and event parameters used in the reported evaluation are summarized in Table 1.

Table 1. Fixed-wing motion and event parameters used in the reported evaluation. Simulator-distance values are converted using the fixed-layer scale of 1000 m per simulator unit. The persistent-proximity threshold is an evaluation termination condition and is distinct from the turning-radius value, even though both are approximately 140 m under the selected configuration.

Parameter	Value
Fixed-wing speed range	20–40 m/s
Steering-command clamp	$\pm 20^\circ$
Fixed-wing length parameters	$l_f = l_r = 0.025$ simulator units
Implied minimum turning radius	approximately 140 m from l_f , l_r , $\delta_{\max}$, and the 1000 m/unit scale
Pickup/contact radius	500 m
Collision termination distance	4 m
Persistent-proximity threshold and duration	140 m for 120 s

Under this fixed-wing configuration, the selected steering clamp and simulator length scale imply a minimum turning radius of approximately 140 m. We set the persistent-proximity threshold to the same one-turn-radius scale, not because it represents a collision distance, but because two fixed-wing UAVs that remain within this range for 120 s have failed to recover separation under the selected motion constraint. Therefore, the vehicle cannot hover or instantly redirect toward an infeasible waypoint; it approaches the command through a bounded forward-flight turn. The state transition is expressed as:

$$x_i^{t+1} = f_{\text{dyn}}(x_i^t, a_i^t),$$

where f_{dyn} denotes the simulator's bounded waypoint-to-motion interpretation. This abstraction supports coordination-level evaluation of fixed-wing retrieval behavior by ensuring that all waypoint decisions are mapped through the same bounded forward-flight dynamics. Detailed controller, wind, and actuator models can be layered onto this waypoint interface in deployment-oriented extensions.

4.4.4. Observation Design

In this study, considering that using only global information or only local information fails to provide sufficient learning performance, an observation structure that combines both local and global information is adopted. However, since directly inputting the entire global map results in excessively high dimensionality, global information is provided as a summarized feature consisting of selected high-value targets.

The observation of each agent is composed as follows:

$$o_i^t = [p_i^t, \theta_i^t, \omega_i^t, m_i^t, q_i^t, z_i^t],$$

- m_i^t : Coordinator-relayed relative positions and distances of the other fixed-wing UAVs
- q_i^t : Local buffer-summary map
- z_i^t : Priority target information

Although each fixed-wing UAV observes the relative positions of the other fixed-wing UAVs, this information is not obtained through direct air-to-air communication among fixed-wing UAVs. Each fixed-wing UAV reports its own state to the base-station coordinator, and the coordinator returns a compact relative-state summary as part of the individual observation.

Before being passed to the policy, position-related quantities are normalized by the map scale, local buffer-summary values are normalized by the cell-buffer maximum, and priority scores are clipped and normalized using the same buffer-summary scale. The cell-buffer

maximum was set to 250 fixed-layer data-amount units and used only for learned-policy input normalization. This normalization is used only by the learned policy and not by the deterministic baselines.

Local Map

The local map contains buffer-summary values for a 3×3 region centered on the agent. It is used for close-range environment perception and immediate movement decisions, contributing to local control stability.

Priority Targets

Instead of directly using the full coordinator-side retrieval information, a score is defined for each cell (u, v) as follows:

$$\text{score}(u, v) = \frac{\hat{D}_t(u, v)}{\max(\|p_i^t - c_{uv}\|_2, r_{\text{floor}})},$$

where $\hat{D}_t(u, v)$ denotes the reported buffered-data amount summarized in cell (u, v) at time t , c_{uv} denotes the center coordinate of cell (u, v) , and r_{floor} is the contact-radius-scale distance floor used to avoid singular or excessively large scores when an agent is already close to a candidate. Based on this score, the top- K cells are selected as priority targets. For each selected target, the policy receives its relative position $(\Delta x, \Delta y)$ and score value. The score combines reported buffer amount and distance, enabling compact retrieval guidance without directly using the full coordinator-side buffer-summary map. In this study, K was set to 4 to retain only a small set of high-priority retrieval cues while keeping the observation space compact. In the integrated evaluation, the selected policy uses an exclusive priority cue so that the two fixed-wing agents are guided by non-overlapping priority entries when such entries are available. This cue provides a stronger and more reproducible collection signal for the learned waypoint policy, while the policy still determines the continuous waypoint action from its local buffer-summary map, priority cue, and coordinator-relayed relative fixed-wing state.

4.4.5. Reward Design

The total reward is defined as follows:

$$r_i^t = r_{\text{data}}^t + r_{\text{space}}^t + r_{\text{close}}^t + r_{\text{time}}^t + r_{\text{bound}}^t$$

The reward design balances data retrieval, inter-agent spacing, and operational constraints. If only the data collection reward r_{data}^t is used, multiple agents may concentrate on the same high-value regions, reducing coverage efficiency and increasing close-proximity risk. To mitigate this, r_{space}^t provides a small shaping reward that encourages agents to maintain separation, while r_{close}^t imposes a stronger close-range penalty on risk-zone entry. This proactive spacing mechanism is particularly important for fixed-wing UAVs because their turning-radius constraints make separation difficult once agents enter a densely packed state. The remaining terms, r_{time}^t and r_{bound}^t , discourage passive behavior and inefficient movement outside the valid data-retrieval area.

Data Collection Reward

A reward proportional to the collected data amount c_i^t is provided. Here, λ_1 is a fixed hyperparameter that controls the relative scale of the data collection reward:

$$r_{\text{data}}^t = \frac{c_i^t}{\lambda_1}.$$

Spacing Reward

For the inter-agent distance $d_{ij}^t = \|p_i^t - p_j^t\|$, a reward is given when the distance increases within a certain range. Here, d_s is a distance threshold hyperparameter used for the spacing reward, and λ_2 is a fixed hyperparameter that controls the relative scale of this term:

$$r_{\text{space}}^t = \lambda_2 \cdot \left(\sum_{j \neq i} [d_{ij}^t \leq d_s] \cdot (d_{ij}^t - d_{ij}^{t-1}) \right).$$

This term is used as an auxiliary shaping signal that encourages separation recovery inside the spacing threshold. Its effect is evaluated empirically through the termination and proximity metrics reported in the standalone and integrated experiments. Note that the indicator is evaluated on the current distance d_{ij}^t . An entry step into the threshold therefore yields a negative term that includes the distance decrease from the previous outside position, whereas the exit step earns nothing because the indicator is zero once $d_{ij}^t > d_s$. Any in-and-out cycle across the boundary thus telescopes to a strictly negative net contribution, so repeated boundary crossing cannot be exploited for positive reward. Consistent with this property, the spacing term accumulated by each agent over an episode was negative in all 30 integrated evaluation episodes of the proposed policy, with a mean accumulated value of -0.293 per episode.

Close-Range Penalty

A penalty is applied when the inter-agent distance falls below a threshold. Here, d_c is a distance threshold hyperparameter used for the close-range penalty, and λ_3 is a fixed hyperparameter that controls the relative scale of this term:

$$r_{\text{close}}^t = -\lambda_3 \cdot \left(\sum_{j \neq i} [d_{ij}^t \leq d_c] \cdot (d_c - d_{ij}^t) \right).$$

Time and Boundary Penalties

The time penalty and boundary penalty are defined as follows. Here, β_1 and β_2 are coefficients that control the fixed magnitude of the time and boundary penalties, respectively:

$$r_{\text{time}}^t = -\beta_1, \quad r_{\text{bound}}^t = -[\text{boundary}] \cdot \beta_2.$$

The reward-related hyperparameters were determined relative to the scale of the data collection reward, which was treated as the primary learning objective. The collected data amount per step is clipped at 5, and the data reward is normalized by $\lambda_1 = 50$. Therefore, the maximum instantaneous data reward is approximately $5/50 = 0.1$. The reward scale is chosen so that data collection remains the primary objective, while spacing and close-range terms provide local coordination shaping. Hard safety outcomes are assessed separately through collision and persistent-proximity termination criteria during evaluation. The spacing reward and the time and boundary penalties were assigned smaller magnitudes, because they were intended to provide auxiliary shaping signals without overwhelming the main objective. The main observation and reward hyperparameters are listed in Table 2.

Table 2. Main hyperparameters used in the proposed observation and reward design. Reward-distance thresholds are expressed in simulator units and are used as shaping thresholds, not as evaluation termination distances.

Item	Value
<i>Observation design</i>	
Local map size (q_i^t)	3×3
Number of priority targets (K)	4
Buffer-summary map grid size	10×10
Priority-score distance floor (r_{floor})	Contact radius, 500 m
Cell-buffer maximum for input normalization	250 fixed-layer data-amount units
<i>Reward design</i>	
Data collection normalization coefficient (λ_1)	50.0
Spacing reward threshold (d_s)	2.0 simulator units
Spacing reward coefficient (λ_2)	0.1
Close-range penalty threshold (d_c)	0.5 simulator units
Close-range penalty coefficient (λ_3)	0.1
Time penalty coefficient (β_1)	0.01
Boundary penalty coefficient (β_2)	0.01

4.4.6. Training Structure

The policy is trained with MAPPO under a parameter-sharing actor, a centralized critic, and per-agent individual rewards. Each component of this configuration addresses a specific difficulty of the fixed-wing setting. The centralized critic accesses the joint state of all agents during training and provides value estimates that are consistent across agents, which reduces the non-stationarity that otherwise arises when each agent treats the others as part of a shifting environment. Parameter sharing across the actor allows the agents to draw on a common pool of experience, which is helpful because the continuous fixed-wing dynamics make per-agent exploration expensive. Individual rewards, rather than a single team reward, preserve the credit signal for each agent's own spacing behavior and data retrieval, so that the spacing-based coordination reward introduced earlier is not diluted by the actions of others. At execution time, the critic is dropped and each fixed-wing UAV acts using its own observation vector. The execution is decentralized at the policy level: fixed-wing UAVs do not negotiate actions with one another and do not exchange payload data or state messages directly. However, each observation may include compact coordinator-relayed metadata, such as retrieval cues and relative fixed-wing state information, derived from the base-station telemetry/control channel.

The proposed RL policy reported in the evaluation was trained with collision-only termination. In evaluation, however, all standalone and integrated runs checked two terminal conditions: collision termination and persistent-proximity termination. Persistent-proximity termination is triggered when fixed-wing UAVs remain inside the predefined risk range longer than the allowed recovery duration, indicating that the pair failed to separate after entering a hazardous close-approach state.

The reported RL evaluations use a fixed checkpoint selected from MAPPO training with training seed 0. All learned fixed-wing RL policies were trained under the same maximum checkpoint-search budget of 300 MAPPO iterations. With 120,000 frames per iteration, this corresponds to a common search budget of 36 million simulator frames under the training implementation. During training, checkpoints were monitored using the training-time mean episode reward. The Proposed RL policy used in the reported evaluation was selected at iteration 99, which achieved the highest training-time mean episode reward within the 300-iteration budget. This checkpoint corresponds to 11.88 million simulator frames.

Subsequent checkpoints remained part of the same search budget but did not exceed the selected checkpoint under the checkpoint-selection criterion. The iteration-99 checkpoint was used as the Proposed RL policy in the standalone and integrated evaluations and as the Proposed setting in the component evaluation; the other learned component variants followed the same checkpoint-selection rule under the same 300-iteration budget. The MAPPO training and checkpoint-selection settings are summarized in Table 3.

Table 3. MAPPO training and checkpoint-selection settings for learned fixed-wing RL policies.

Item	Value
Frames per iteration	120,000
Maximum checkpoint-search iterations	300
Maximum checkpoint-search frames	36,000,000
Rollout length	4000 steps
Scenario-length rollouts per iteration	30
Maximum scenario-length rollouts	9000
Epochs per iteration	3
Mini-batch size	12,000
Learning rate	1×10^{-3}
Discount factor (γ)	0.995
GAE parameter (λ)	0.97
Training seed	0
Checkpoint-selection criterion	Highest training-time mean episode reward within the 300-iteration budget
Selected checkpoint for Proposed RL	Iteration 99
Frames to selected Proposed RL checkpoint	11,880,000
Scenario-length rollouts to selected Proposed RL checkpoint	2970

5. Component-Level Experimental Analysis

This section characterizes the two components that enable the proposed coordination framework. The lower-layer experiment evaluates whether clustered replanning can support repeated route updates as the active target set grows. The upper-layer standalone experiment isolates fixed-wing retrieval behavior before coupling it with the lower-layer routing process. All component-level and integrated evaluations reported in Sections 5 and 6 were conducted on an Ubuntu 22.04 workstation with an Intel Core i9-14900K CPU, 128 GB RAM, and an NVIDIA RTX 4090 GPU, using Python 3.10.19 in a conda environment with PyTorch 2.9.1+cu128, TorchRL 0.10.1, VMAS 1.5.2, and OR-Tools 9.14.6206. All reported computation times and wall-clock measurements refer to this environment.

5.1. Multirotor Route Generation Efficiency

To evaluate the efficiency of the clustering-based route generation method in large-scale node environments, we analyzed its computation time from two perspectives. First, we compared the computation time of the clustering-based route generation method with that of the TSP-based route generation method. Second, we measured how the computation time of the clustering-based method changes as the number of targets increases. The experiment was conducted by varying the number of targets from 50 to 600 in increments of 50, and the average computation time was obtained by repeating each condition 20 times. In addition, to reflect the increase in problem scale more realistically, the number of UAVs was also increased proportionally by assigning one UAV for every 30 targets. Through this setup, we aimed to quantitatively examine both the relative computational advantage of clustering-based partitioning and its scalability as the number of nodes increases.

As shown in Figure 3a, the clustering-based route generation method consistently required less computation time than the TSP-based route-generation proxy across the tested target ranges. This latency test uses a TSP-style routing proxy to isolate the effect of clustering on route-generation time; the full lower-layer planner uses the CVRP formulation summarized in Algorithm 1. This indicates that partitioning the active targets before route generation can effectively reduce the computational burden, compared with directly solving a route over the entire active target set.

Figure 3b shows that the computation time of the clustering-based route generation method increased in a relatively gradual and near-linear manner as the number of targets increased. Even under the 600-target condition, route generation was completed within approximately 1.3 s on average. The latency experiment in Figure 3 evaluates the computational benefit of decomposing the lower-layer route-generation problem. Because fast computation alone is not sufficient if collection quality deteriorates, Figure 4 and Table 4 separately evaluate the effect of the proposed age- and cooldown-aware objective on collection, freshness, redundant revisits, and collection efficiency.

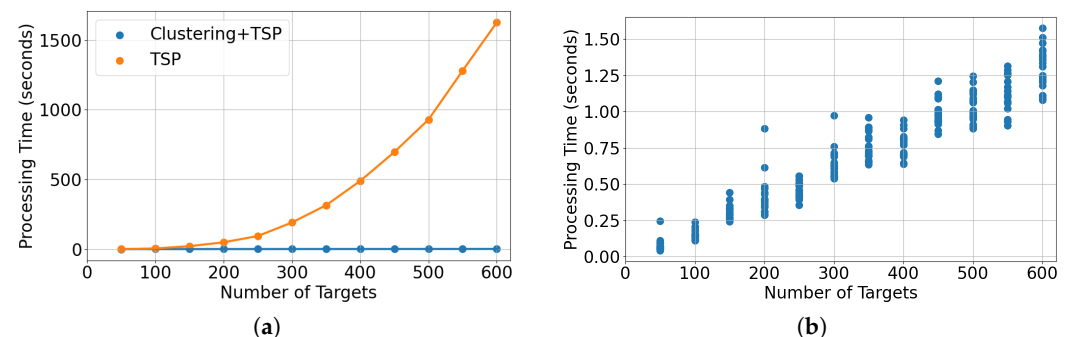


Figure 3. Computation time analysis of the clustering-based route generation method. (a) Comparison of computation time between clustering-based route generation and TSP-based route generation. (b) Computation time of the clustering-based route generation method as the number of targets increases.

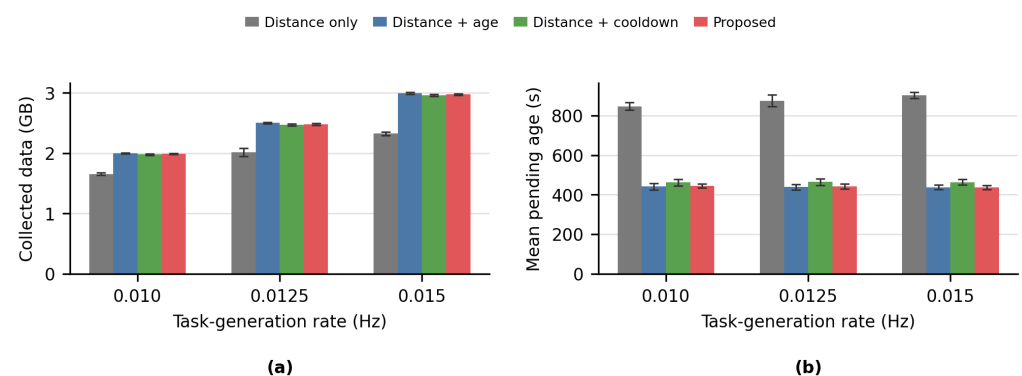


Figure 4. Lower-layer routing objective effects under lower-layer objective evaluation with a fixed greedy retrieval placeholder. Error bars denote standard deviation over 10 seeds. (a) Collected data for the four routing objectives across the three task-generation rates. (b) Mean pending age for the four routing objectives across the three task-generation rates.

Table 4. Additional lower-layer routing objective diagnostics that complement Figure 4. Values are mean $\pm$ standard deviation over 10 seeds with a 4800 s horizon and a greedy fixed-wing placeholder. In the cooldown-aware and proposed objectives, the cooldown term is implemented as a soft routing-cost penalty. Stale ratio and redundant revisits are reported as percentages, and collection/distance as bytes per meter.

Rate (Hz)	Objective	Stale Ratio	Redundant Revisits	Collection/Distance (Bytes/m)
0.01	Distance only	34.0 $\pm$ 0.9	64.1 $\pm$ 1.3	1129.0 $\pm$ 15.0
0.01	Age-aware	12.8 $\pm$ 1.1	36.5 $\pm$ 2.3	1358.7 $\pm$ 6.2
0.01	Cooldown-aware	13.0 $\pm$ 1.0	20.0 $\pm$ 1.8	1343.9 $\pm$ 8.6
0.01	Proposed	12.5 $\pm$ 0.8	18.3 $\pm$ 2.5	1347.3 $\pm$ 6.8
0.0125	Distance only	35.3 $\pm$ 1.3	65.8 $\pm$ 2.1	1372.9 $\pm$ 47.1
0.0125	Age-aware	12.6 $\pm$ 0.9	36.9 $\pm$ 2.1	1698.5 $\pm$ 7.1
0.0125	Cooldown-aware	13.1 $\pm$ 1.0	20.6 $\pm$ 1.8	1676.3 $\pm$ 11.9
0.0125	Proposed	12.4 $\pm$ 0.9	20.1 $\pm$ 2.2	1681.0 $\pm$ 11.7
0.015	Distance only	36.6 $\pm$ 0.6	68.0 $\pm$ 1.0	1585.3 $\pm$ 21.1
0.015	Age-aware	12.6 $\pm$ 0.8	37.2 $\pm$ 2.3	2032.1 $\pm$ 13.3
0.015	Cooldown-aware	13.1 $\pm$ 0.9	20.7 $\pm$ 1.8	2007.8 $\pm$ 13.9
0.015	Proposed	12.1 $\pm$ 0.8	20.8 $\pm$ 2.4	2013.8 $\pm$ 12.7

This evaluation keeps the upper fixed-wing layer as a greedy retrieval placeholder and compares four routing objectives in the multirotor layer: distance-only routing, an age-aware objective, a cooldown-aware objective, and the proposed objective, which combines age-weighted distance with a separate cooldown penalty. Each sensor node generates data items according to the per-node process in Table 5; the rates 0.01, 0.0125, and 0.015 Hz correspond to approximately 2.39, 2.98, and 3.58 GB of generated data over the 4800 s scenario, respectively. Figure 4 reports the primary throughput and mean-freshness outcomes, while Table 4 keeps the additional diagnostics that complement Figure 4.

Table 5. Lower-layer implementation parameters used in the reported integrated evaluation. The data-item generation rate λ is varied in the overall experiment; the remaining values are fixed unless otherwise noted.

Parameter	Value
Clustering input	All deployed ground nodes $\mathcal{N}$
Routed demand input	Nodes with nonempty pending-task sets $\mathcal{N}_i^{\text{act}}$ inside each cluster
Lloyd iterations L	10
Replanning period T_{rp}	60 s
Data-item generation process	Per-node Poisson process, implemented by a per-step Bernoulli draw with probability $1 - \exp(-\lambda \Delta t)$
Data-item size s_{r}	50,000 bytes
Multirotor speed	10 m/s
Per-multirotor buffer/routing capacity Q	5,000,000,000 bytes
Route-time bound $T_{\text{max}}^{(k)}$	2400 s minus current flight time, with a 60 s minimum planning guard
Node service time	5 s
Age coefficient α	0.005 s $^{-1}$
Cooldown penalty ρ	1000 m-equivalent cost units
Cooldown window Δ_{cool}	900 s
OR-Tools first solution strategy	PATH_CHEAPEST_ARC
OR-Tools per-call time limit	3000 ms
Per-node drop penalty	100,000
Contact radius for payload pickup	500 m

Figure 4 shows the main effect of adding temporal terms to the lower-layer route cost. Distance-only routing leaves noticeably lower collected data and much older pending tasks, whereas the age-aware, cooldown-aware and proposed objectives maintain high collection while sharply reducing mean pending age. Table 4 adds complementary diagnostics for stale tasks, repeated short-interval visits, and collection efficiency. The proposed objective gives the lowest stale ratio at all three rates and reduces redundant revisits to a level comparable to the cooldown-aware objective while keeping collection per distance close to the

single-term temporal objectives. These results indicate that age weighting and cooldown penalization play complementary roles: the former prioritizes old pending data, while the latter suppresses repeated short-interval visits. Together with the route-generation latency result, the lower-layer analysis suggests that clustering-based decomposition supports efficient periodic replanning, while the age-weighted and cooldown-aware routing costs improve freshness, reduce redundant revisits, and preserve or improve collection efficiency relative to distance-only routing. The proposed objective is therefore used as a balanced lower-layer routing rule before the fixed-wing retrieval policy is evaluated in the integrated system.

5.2. Standalone Fixed-Wing Retrieval Comparison

Before the integrated two-layer evaluation, we isolate the fixed-wing retrieval problem and compare the learned waypoint policy with two scale-matched greedy strategies. This standalone evaluation tests whether the learned policy can use the same compact retrieval information to achieve competitive pickup behavior while avoiding termination events before lower-layer coupling is introduced. Greedy Exclusive is treated as a strong collection baseline, while Greedy Independent provides a contrast for non-exclusive target selection. The Proposed RL row follows the MAPPO training and checkpoint-selection protocol defined in Table 3.

Table 6 shows that the proposed RL policy and the scale-matched Greedy Exclusive baseline have comparable collection outcomes and both complete the standalone evaluation without termination events. Greedy Independent collects less data and terminates frequently, illustrating why non-exclusive target selection is a poor contrast case for this setting. This standalone fixed-wing retrieval result establishes the upper-layer comparison frame used in the integrated system-level evaluation.

Table 6. Standalone fixed-wing retrieval evaluation for the upper fixed-wing layer. Values are mean $\pm$ standard deviation over 10 evaluation seeds unless otherwise stated. Collection ratio is the evaluator-reported collected-data ratio for the standalone episode. Termination events count evaluation seeds that ended by collision termination or persistent-proximity termination.

Strategy	Collection Ratio (Collected/ Available)	Collected Amount	Remaining Data	Termination Events
Proposed RL	1.0000 $\pm$ 0.0000	6378.2 $\pm$ 217.9	0.0035 $\pm$ 0.0111	0
Greedy Exclusive	0.9994 $\pm$ 0.0010	6255.0 $\pm$ 535.0	4.1 $\pm$ 6.5	0
Greedy Independent	0.9239 $\pm$ 0.0683	5911.3 $\pm$ 440.5	494.3 $\pm$ 443.6	8

6. Overall Experimental Results

We evaluated the integrated two-layer system by fixing the lower-layer multirotor operation and varying only the upper-layer fixed-wing data-retrieval strategy. This experiment examines whether the selected fixed-wing policy remains effective when combined with the lower-layer routing system. The lower-layer objective ablation in the previous section provides component-level evidence for the multirotor routing rule, while the integrated experiment measures the resulting two-layer delivery behavior.

For a fair comparison, all upper-layer methods were evaluated in the same VMAS-based environment [51], using the same fixed-wing motion settings, the same bounded waypoint-to-motion conversion, and the same lower-layer multirotor operation. Thus, the comparison focuses on the high-level waypoint or target selection strategy of the fixed-wing UAVs, rather than differences in vehicle dynamics, motion-conversion implementation, or lower-layer routing behavior.

The primary integrated comparison includes four upper-layer retrieval strategies:

- *Proposed RL*: The selected learned waypoint policy uses the local-priority observation and the exclusive priority cue, while the policy determines continuous waypoint actions through the bounded waypoint-to-motion conversion layer.
- *Greedy Exclusive*: Priority-target selections are made exclusive across agents where possible, providing a strong collection-oriented deterministic baseline.
- *Receding Horizon*: A joint finite-horizon optimization baseline evaluates candidate fixed-wing retrieval sequences using predicted collection, travel cost, duplicate-target and repeated-cell penalties, and spacing-related safety terms for collision avoidance and for discouraging prolonged close-range residence under the persistent-proximity criterion. It uses a three-step planning horizon, a five-step replanning interval, top-8 candidate retrieval targets per agent, and a beam width of 4; only the first waypoint command from the selected sequence is executed before the next replanning update.
- *Ergodic Search*: A coverage-density optimization baseline generates conservative retrieval behavior by encouraging trajectory visitation to match the coordinator-derived buffered-data density while accounting for motion smoothness and separation costs. The density is derived from the coordinator-provided buffered-data map and moving multirotor buffer context. The evaluated controller uses a trajectory horizon of 125 steps with three optimization iterations per update, and the resulting command is passed through the same fixed-wing waypoint-to-motion conversion used by the other strategies.

The proposed RL policy, Greedy Exclusive, and Receding Horizon methods use priority targets computed from reported buffer amount and distance. Greedy Exclusive serves as a high-throughput target-selection baseline that directly converts exclusive priority targets into waypoint commands. It does not explicitly optimize spacing or persistent-proximity recovery at the high-level decision layer, although it uses the same bounded fixed-wing motion module and evaluator as the other methods. The proposed RL policy addresses a different coordination role: it receives the same type of compact priority cue, together with local buffer information and coordinator-relayed relative fixed-wing states, and converts them into bounded waypoint actions under spacing-aware motion constraints. Receding Horizon provides a conventional optimization-based benchmark for the same retrieval interface, whereas the Ergodic Search controller represents a more conservative coverage-density optimization style. The comparison therefore evaluates whether learned fixed-wing maneuvering can preserve collection performance while maintaining favorable termination behavior under the same lower-layer workload. The primary performance comparison is not based on runtime alone; nevertheless, we also report wall-clock evaluation time as a supplementary check of whether the upper-layer decision strategies fit within the 1 s simulation-step interval. All strategies are evaluated under the same lower-layer operation, same modified VMAS fixed-wing motion update, and same bounded waypoint-to-motion conversion. The baselines differ only in the high-level retrieval decision used to generate the next waypoint command. The non-exclusive Greedy Independent variant was also evaluated as a diagnostic control, but all 30 episodes terminated early; it is therefore excluded from the primary comparison rather than treated as a balanced collection baseline.

Each primary strategy was evaluated over 10 evaluation seeds for each reported data-generation rate, producing 120 primary integrated episodes in total. The reported rates are 0.01, 0.0125, and 0.015 Hz, which cover the selected policy's training-rate neighborhood. The evaluation length was set to 4800 steps, where one step corresponds to one second. Unless otherwise noted, collected amount in the integrated evaluation is reported in fixed-layer data-amount units after data-unit conversion; under the reported data-unit setting, one fixed-layer data-amount unit corresponds to 250,000 bytes, or five generated 50,000-byte

data items. This duration is twice the maximum route time of the lower-layer multirotor operation, which is set to 2400 s. Therefore, the experiment observes not only the behavior within a single lower-layer route cycle, but also the accumulated effect of the upper-layer fixed-wing data-retrieval strategy over two lower-layer operation cycles. As described in the observation design, the cell-buffer maximum is used as an input-normalization constant for the learned policy. Greedy Exclusive and the optimization baselines compute their target ordering or objective terms from the current coordinator-provided retrieval state, and do not use this value as a learned-model normalization parameter. Therefore, all compared methods share the same lower-layer data generation, routing state, vehicle dynamics, and evaluation episodes. The integrated scenarios use 1000 nodes, 32 multirotor UAVs, and 2 fixed-wing UAVs in a $10,000 \times 10,000$ m area. The 32-multirotor and 2-fixed-wing setting is used as a controlled retrieval-bottleneck configuration: the lower layer generates a spatially distributed moving-buffer field, while the upper layer has limited retrieval capacity. This setting is intended to stress the fixed-wing retrieval strategy rather than to exhaustively characterize fleet-ratio scaling.

6.1. Component Evaluation of Proposed RL

To examine whether the selected Proposed RL setting depends on a single design factor, we further evaluated component-level Proposed RL variants in the integrated two-layer scenario. For each component study, we retrained the corresponding Proposed RL variant while changing only one design factor from the selected Proposed RL configuration. Each trained variant was then evaluated in the same integrated two-layer UAV scenario for ten evaluation seeds, 0–9, at $\lambda = 0.0125$ Hz over a 4800-step horizon, using the same collision and persistent-proximity termination criteria. In the following tables, termination events are integer counts over the ten seeds, steps within 140 m are totals over the ten seeds, minimum observed distance denotes the minimum fixed-wing center-to-center distance observed over all ten evaluation seeds, and “(Proposed)” marks the selected Proposed RL setting. These component tables are intended as diagnostic sensitivity checks for the selected policy configuration; the primary rate-specific comparison with standard deviations is reported in Table 7. All learned RL variants in the component evaluation followed the same checkpoint-selection protocol. Each variant was trained with training seed 0 under the common maximum budget of 300 MAPPO iterations, and the checkpoint with the highest training-time mean episode reward within that budget was selected. The selected checkpoint of each variant was then evaluated under the same integrated evaluator and evaluation seeds. This protocol gives each learned variant the same training budget and compares selected-checkpoint performance under a common selection rule. The selected iteration may differ across variants; the common 300-iteration budget was used to give each variant the same checkpoint-search opportunity before evaluation.

Table 8 shows that the local-only observation avoided termination, but its collected amount and minimum-distance margin remained below those of the selected cue configuration. Adding the exclusive priority cue to the local observation increased collection while maintaining termination-free behavior. This comparison indicates that compact priority guidance with exclusive assignment is more suitable for the integrated workload than the local buffer-summary map alone, without treating the local-only variant as a failed safety case.

Table 7. Rate-specific primary integrated evaluation summary. Each strategy-rate pair aggregates 10 evaluation seeds; the four primary strategies over three data-generation rates therefore give 120 primary integrated evaluation episodes in total. Collected amount and mean minimum distance are reported as mean $\pm$ standard deviation over 10 seeds. Steps within 140 m are totals over the 10 seeds for each strategy-rate pair. Termination events are integer counts over the 10 seeds, without denominator notation. All methods use the same environment, lower-layer routing, data-unit configuration, fixed-wing motion module, motion settings, and evaluator, and differ only in the fixed-wing waypoint decision strategy.

λ (Hz)	Strategy	Collected Amount	Termination Events	Steps Within 140 m	Mean Minimum Distance (m)
0.01	Proposed RL	3825.6 $\pm$ 71.5	0	161	60.9 $\pm$ 43.5
0.01	Greedy Exclusive	3812.3 $\pm$ 55.0	0	125	109.0 $\pm$ 83.0
0.01	Receding Horizon	3201.1 $\pm$ 364.9	0	111	108.2 $\pm$ 67.4
0.01	Ergodic Search	2848.9 $\pm$ 138.6	0	6	455.5 $\pm$ 298.4
0.0125	Proposed RL	4650.1 $\pm$ 126.5	0	294	85.1 $\pm$ 64.9
0.0125	Greedy Exclusive	4401.1 $\pm$ 902.9	1	195	70.6 $\pm$ 61.3
0.0125	Receding Horizon	3814.8 $\pm$ 380.0	0	124	98.7 $\pm$ 72.8
0.0125	Ergodic Search	3246.0 $\pm$ 351.9	0	11	572.4 $\pm$ 303.6
0.015	Proposed RL	5445.7 $\pm$ 167.0	0	208	55.2 $\pm$ 39.0
0.015	Greedy Exclusive	5501.7 $\pm$ 161.9	0	132	78.4 $\pm$ 57.5
0.015	Receding Horizon	4502.5 $\pm$ 238.6	0	95	116.5 $\pm$ 109.1
0.015	Ergodic Search	3585.3 $\pm$ 346.8	0	0	621.2 $\pm$ 253.4

Table 8. Evaluation of the priority-aware assignment cue in Proposed RL variants.

Priority Cue	Mean Collected Data	Terminations	Steps Within 140 m	Minimum Observed Distance (m)
Local-only observation	3302.7	0	202	6.3
Local + exclusive priority cue (Proposed)	4650.1	0	294	29.8

Table 9 indicates that the smallest priority set was unstable in the integrated scenario, with seven termination events and a much lower mean collected amount. In contrast, $K = 2$, $K = 4$, $K = 6$, and $K = 8$ all remained termination-free and collected more than 4500 on average. The selected $K = 4$ setting had the highest mean collected amount in this group, whereas $K = 6$ had fewer steps within 140 m and a larger minimum-distance tail. Thus, the result supports a moderate priority-cell range rather than a claim that only $K = 4$ can operate.

Table 9. Evaluation of the priority-cell parameter in Proposed RL variants.

Priority-Cell K	Mean Collected Data	Terminations	Steps Within 140 m	Minimum Observed Distance (m)
1	1506.8	7	825	3.0
2	4507.3	0	194	12.0
4 (Proposed)	4650.1	0	294	29.8
6	4604.8	0	163	34.3
8	4603.4	0	193	18.7

Table 10 separates the training signal from the evaluation rule. Although the no-termination-signal variant also completed all ten evaluation seeds without termination, it collected less than the selected collision-signal setting. The proximity-signal variant had the highest mean collected amount and fewer steps within 140 m, but its minimum distance dropped to 4.8 m. The collision-plus-proximity variant produced two termination events, showing that adding all safety-related termination signals during training does not monotonically improve the integrated outcome.

Table 10. Evaluation of the training termination signal in Proposed RL variants. All rows were evaluated with the same collision and persistent-proximity termination criteria.

Termination Signal	Mean Collected Data	Terminations	Steps Within 140 m	Minimum Observed Distance (m)
No termination signal	4566.4	0	209	14.1
Collision signal (Proposed)	4650.1	0	294	29.8
Proximity signal	4688.6	0	125	4.8
Collision + proximity signals	4461.1	2	239	3.6

As shown in Table 11, all spacing-reward-scale variants were termination-free, but the selected scale had the highest mean collected amount and the largest minimum distance in this group. Turning off the spacing reward or using a lower scale reduced collection, while the higher scale also reduced collection and increased the accumulated residence within 140 m. The spacing reward therefore acts as a trade-off factor between collection pressure and close-range residence, rather than as an isolated guarantee of safety.

Table 11. Evaluation of the spacing reward scale in Proposed RL variants.

Spacing Reward Scale	Mean Collected Data	Terminations	Steps Within 140 m	Minimum Observed Distance (m)
0.0	4475.3	0	156	6.0
0.05	4475.4	0	216	18.3
0.1 (Proposed)	4650.1	0	294	29.8
0.2	4540.2	0	350	10.8

Table 12 shows that the selected spacing reward distance achieved the highest collection and avoided termination in this component group. The smaller and larger distances both produced collision termination events and more aggressive minimum-distance tails. This result suggests that spacing-distance tuning alone is not sufficient to ensure integrated safety behavior.

Table 12. Evaluation of the spacing reward distance in Proposed RL variants.

Spacing Reward Distance	Mean Collected Data	Terminations	Steps Within 140 m	Minimum Observed Distance (m)
1.0	4340.4	1	200	2.6
2.0 (Proposed)	4650.1	0	294	29.8
4.0	3732.7	2	248	2.4

Table 13 shows a milder trade-off. All tested penalty-scale variants completed the ten integrated episodes without termination. The off and low-scale settings produced similar collection values but lower minimum-distance tails than the selected setting. The high-scale setting improved the minimum-distance tail relative to those two settings, but its collected amount was lower than that of the selected setting.

Table 13. Evaluation of the too-close penalty scale in Proposed RL variants.

Too-Close Penalty Scale	Mean Collected Data	Terminations	Steps Within 140 m	Minimum Observed Distance (m)
0.0	4611.3	0	151	10.7
0.05	4614.3	0	170	7.0
0.1 (Proposed)	4650.1	0	294	29.8
0.2	4572.0	0	148	24.5

Table 14 indicates that the selected too-close distance had the highest mean collected amount while remaining termination-free. The 0.75 setting also avoided termination, but with lower collection and a smaller minimum-distance tail. The 1.0 setting had the fewest steps within 140 m, but also reduced collection. The 0.25 setting produced one persistent-proximity termination, so lowering the threshold should not be interpreted as a uniformly safer choice.

Table 14. Evaluation of the too-close distance in Proposed RL variants.

Too-Close Distance	Mean Collected Data	Terminations	Steps Within 140 m	Minimum Observed Distance (m)
0.25	4377.3	1	237	31.8
0.5 (Proposed)	4650.1	0	294	29.8
0.75	4485.0	0	153	9.2
1.0	4334.7	0	74	8.8

Table 15 shows that the selected data collection normalization coefficient had the highest collection and completed all ten episodes without termination. Increasing the coefficient to 100.0 remained termination-free but reduced both collection and the minimum-distance tail, while the 25.0 coefficient produced one collision termination and the smallest minimum-distance value. This component therefore affects the collection-safety trade-off rather than providing a monotonic improvement through larger or smaller data-reward normalization factors.

Table 15. Evaluation of the data collection normalization coefficient (λ_1) in Proposed RL variants.

Data Collection Normalization Coefficient (λ_1)	Mean Collected Data	Terminations	Steps Within 140 m	Minimum Observed Distance (m)
25.0	4237.3	1	121	1.2
50.0 (Proposed)	4650.1	0	294	29.8
100.0	4496.6	0	229	11.3

Overall, the component study shows that the selected Proposed RL configuration is not a single-factor artifact. The exclusive priority cue improved collection relative to the local-only observation while preserving termination-free behavior, and moderate-to-large priority-cell settings remained termination-free. The termination, spacing, too-close, and data-reward normalization factors affected collection, termination behavior, close-range residence, and minimum-distance tails in different ways. Some variants reduced near-distance residence but also reduced collection or produced more aggressive minimum-distance tails. The selected Proposed RL setting was therefore retained because it preserved high collection in the integrated scenario while avoiding collision and persistent-proximity termination events under the common evaluator.

6.2. Experimental Results

Figure 5 and Table 7 summarize the primary integrated comparison across the data-generation-rate sweep, and Table 16 reports the corresponding evaluation-side wall-clock time. The Proposed RL entries follow the selected-checkpoint protocol defined in Table 3. Panel (a) shows that Proposed RL has the highest model-family mean collected amount, 4640.5, followed by Greedy Exclusive at 4571.7, Receding Horizon at 3839.5, and Ergodic Search at 3226.7. At the reference workload of 0.0125 Hz, the Proposed RL policy collected 4650.1, compared with 4401.1 for Greedy Exclusive, 3814.8 for Receding Horizon, and 3246.0 for Ergodic Search. At the lower rate of 0.01 Hz, the proposed policy and Greedy Exclusive remained nearly tied in collection, while Receding Horizon collected less than those two

collection-oriented strategies but completed all 10 episodes without termination. At the higher rate of 0.015 Hz, all primary strategies increased their collected amount, and Greedy Exclusive produced the highest single-rate mean, 5501.7, with the proposed policy close behind at 5445.7.

Table 16. Supplementary wall-clock evaluation time for the primary integrated strategies. Each row aggregates the same 30 integrated evaluation episodes used for the model-family comparison, consisting of 10 seeds at each of the three data-generation rates. Mean step time divides the mean evaluation time by the 4800-step horizon and represents evaluation-side wall-clock time, including environment stepping, lower-layer operation, and runner/logging overhead, rather than pure policy-inference latency.

Strategy	Integrated Episodes	Mean Evaluation Time (s)	Mean Step Time (s/Step)
Proposed RL	30	372.1	0.0775
Greedy Exclusive	30	334.8	0.0698
Receding Horizon	30	335.9	0.0700
Ergodic Search	30	8281.8	1.7254

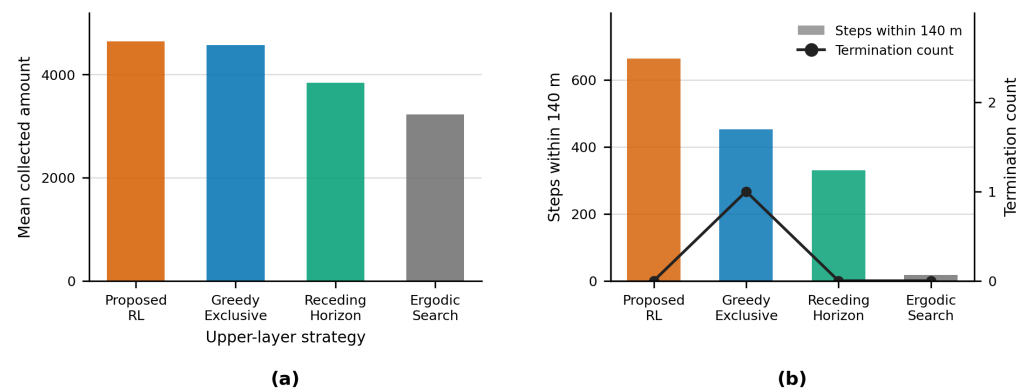


Figure 5. Primary integrated two-layer evaluation summary. Panel (a) is a bar chart comparing the model-family mean collected amount over 30 integrated evaluation episodes for each upper-layer strategy; the 30 episodes consist of 10 evaluation seeds at each of the three data-generation rates, 0.01, 0.0125, and 0.015 Hz. Panel (b) summarizes safety outcomes aggregated over the three rates: bars show the total number of steps in which the two fixed-wing UAVs were within 140 m, and the line shows termination-event counts. Termination events include collision termination and persistent-proximity termination reported by the integrated evaluator.

The safety outcomes clarify the collection-safety trade-off across the same rate sweep. These results should not be interpreted as showing that the proposed policy minimizes all close-proximity exposure. In fact, Proposed RL records more steps within the 140 m recovery-zone metric than the more conservative optimization baselines. This indicates that the learned policy can enter recoverable close-approach states while pursuing high-value pickups. However, none of these close-approach periods escalated into collision or persistent-proximity termination under the common evaluator and evaluated scenarios. Greedy Exclusive achieves strong collection through direct priority following, but produced one termination event at the reference workload. Receding Horizon provides a finite-horizon optimization baseline with explicit collection and safety terms; however, under the selected candidate set, horizon, and replanning interval, it behaves more conservatively and leaves more residual buffered data than the learned policy. Ergodic Search represents a conservative coverage-density optimization style that preserves separation, but its objective is less aligned with the immediate retrieval pressure of moving multirotor buffers under rapidly accumulating data. Because the reported values are averaged over 10 seeds per

condition, the comparison is interpreted as an operating-regime characterization rather than as a claim of statistical dominance on every metric. Therefore, the key outcome of Proposed RL is high-throughput, termination-free retrieval under the evaluated fixed-wing motion and safety evaluator, rather than uniformly greater inter-agent separation.

The runtime results in Table 16 show that Proposed RL required 0.0775 s per step on average, which is well below the 1 s simulation-step interval and close to the Greedy Exclusive and Receding Horizon baselines. Thus, the proposed policy preserves the practical runtime profile of lightweight waypoint-selection strategies in the common evaluator, while Ergodic Search is substantially more expensive because it performs coverage-density optimization during evaluation. Within the short-runtime group, Proposed RL achieved the highest 30-episode mean collected amount and completed the reported rate sweep without collision or persistent-proximity termination, whereas Greedy Exclusive produced one termination event and Receding Horizon collected less data. These results indicate that Proposed RL provides a favorable integrated collection-termination trade-off without making the primary comparison a runtime-only claim.

Figure 6 complements the endpoint statistics by showing the remaining-buffer trajectory over the full integrated horizon. Because the 4800-step evaluation covers two 2400-s lower-layer operation cycles, the curves show both the accumulation of newly buffered data and the ability of each fixed-wing strategy to clear that buffer over repeated cycles. Proposed RL and Greedy Exclusive keep the remaining-buffer envelope substantially below Receding Horizon and Ergodic Search, which is consistent with their higher collected amounts in Figure 5 and Table 7. The proposed policy should therefore be interpreted as maintaining high retrieval throughput while avoiding collision and persistent-proximity termination events, rather than as dominating Greedy Exclusive at every time step. Receding Horizon and Ergodic Search leave larger residual buffered data, reflecting the throughput cost of their more conservative or optimization-driven retrieval behavior under the same lower-layer workload.

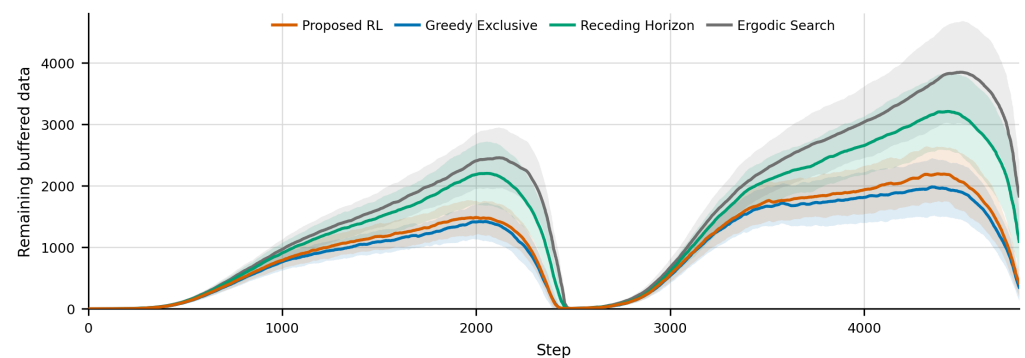


Figure 6. Remaining buffered data over the 4800-step integrated evaluation horizon for Proposed RL, Greedy Exclusive, Receding Horizon, and Ergodic Search. Lines show the mean remaining buffered data across the available rollout samples at each time step, and shaded bands show ± 1 standard deviation. Each strategy starts from 30 planned integrated evaluation episodes, consisting of 10 evaluation seeds at each of the three data-generation rates, 0.01, 0.0125, and 0.015 Hz; after the terminated Greedy Exclusive rollout, its later time steps are averaged over the remaining 29 samples. The horizon covers two 2400-s lower-layer operation cycles, so the plot visualizes how each fixed-wing retrieval strategy clears or leaves buffered data over repeated lower-layer route cycles. In particular, the sharp decrease near the 2400-step boundary occurs when the lower-layer multirotor routes reach their maximum route-time limit and the simulator applies the route-cycle reset/return-to-rendezvous update. All strategies use the same lower-layer operation, fixed-wing dynamics, waypoint-to-motion conversion, and evaluator; they differ only in the fixed-wing waypoint decision strategy.

7. Conclusions

In this paper, we proposed a two-layer UAV coordination framework for fast and efficient data collection in large-scale dynamic environments. Our proposed framework assigns each platform to the role for which it is best suited. In the lower layer, multirotor UAVs perform proximity-based local sensing and buffer the collected payload, while in the upper layer, a small number of fixed-wing UAVs retrieve the buffered payload over a wide area and deliver it to the base-station coordinator. The two layers are coupled through a metadata-assisted data-mule interface, in which the coordinator converts reported multirotor buffer metadata into compact summaries and retrieval cues used by the fixed-wing policy. This separation targets operating regimes in which maintaining continuous long-range connectivity from every sensor node is not the intended operating mode, and in which sensing, buffering, and long-range retrieval are operationally separated.

For the lower layer, the proposed replanning method combines warm-started clustering with a per-cluster capacitated vehicle routing problem whose cost integrates an age-weighted distance term and a separate cooldown penalty. The component-level analysis shows that clustering-based decomposition keeps route generation tractable as the active target set grows, completing route generation for up to 600 targets within approximately 1.3 s on average, well within the replanning interval used in the integrated evaluation. The objective study further shows that age weighting and cooldown penalization play complementary roles, improving data freshness and reducing redundant short-interval revisits relative to distance-only routing while preserving collection efficiency.

For the upper layer, the learned fixed-wing retrieval policy converts a local buffer-summary map and compact priority cues into bounded waypoint actions under fixed-wing motion and spacing constraints. Across the reported data-generation-rate sweep, the policy attains the highest model-family mean collected amount among the evaluated strategies and clears the buffered-data envelope comparably to a strong exclusive greedy baseline, while recording no collision or persistent-proximity termination events under the common evaluator. The policy also preserves a practical per-step runtime well below the simulation-step interval, which indicates that the learned coordination does not come at the cost of evaluation-time feasibility.

Taken together, these results indicate that the contribution of the framework lies not in the individual use of hierarchical structure, heterogeneous platforms, clustered routing, or reinforcement learning, each of which is established in the literature, but in their integration into a single coupled routing-retrieval design under platform-specific motion constraints. The component-level study supports this view, because naive configurations of the same architecture fail to preserve safety or collection performance. The framework is therefore positioned as a platform-aware coordination-layer design that addresses the joint behavior of moving multirotor buffers and motion-constrained fixed-wing retrieval, rather than optimizing routing, relay communication, or learned motion control in isolation.

Several limitations remain and motivate future work. The present study evaluates the framework at the coordination level, representing packet-level communication effects and high-fidelity autopilot dynamics through coordination-level abstractions. Building on the contact-pickup and metadata-exchange interface defined in this paper, deployment-oriented extensions can incorporate finite-rate payload transfer, telemetry delay, packet loss, link failure, synchronization delay, and localization uncertainty, as well as broader fleet-ratio and environmental conditions, while preserving the routing-retrieval structure studied here. Validating the framework under these more detailed communication and sensing conditions is a natural next step toward bridging the coordination-layer design and full deployment.

Author Contributions: conceptualization, S.K. and S.Y.; methodology, S.K. and S.Y.; software, S.K. and S.Y.; validation, S.K., S.Y., and W.J.; investigation, S.K., M.K., U.J., and W.J.; writing—original draft preparation, S.K., S.Y., M.K., and U.J.; writing—review and editing, S.K., S.Y., W.J. and H.K.; visualization, S.K. and S.Y.; supervision, H.K.; project administration, H.K.; funding acquisition, H.K.; All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Korea Research Institute for defense Technology planning and advancement (KRIT) - Grant funded by Defense Acquisition Program Administration (DAPA) (KRIT-CT-23-041).

Data Availability Statement: The simulation data and aggregation tables supporting the reported results are available from the corresponding author upon reasonable request.

Acknowledgments: During the preparation of this manuscript, the authors used ChatGPT (GPT-5.5) by OpenAI to revise the grammar of the manuscript. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

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

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
IoT	Internet of Things
VRP	Vehicle Routing Problem
CVRP	Capacitated Vehicle Routing Problem
TSP	Traveling Salesman Problem
MAPF	Multi-Agent Path Finding
RL	Reinforcement Learning
MARL	Multi-Agent Reinforcement Learning
DRL	Deep Reinforcement Learning
PPO	Proximal Policy Optimization
MAPPO	Multi-Agent Proximal Policy Optimization
MADDPG	Multi-Agent Deep Deterministic Policy Gradient
CTDE	Centralized Training, Decentralized Execution
GAE	Generalized Advantage Estimation
NTN	Non-Terrestrial Network

References

1. Shakhathreh, H.; Sawalmeh, A.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I.; Othman, N.S.; Khreishah, A.; Guizani, M. Unmanned Aerial Vehicles: A Survey on Civil Applications and Key Research Challenges. *IEEE Access* **2019**, *7*, 48572–48634. <https://doi.org/10.1109/ACCESS.2019.2909530>.
2. Yu, Y.; Lee, S. Efficient Multi-UAV Path Planning for Collaborative Area Search Operations. *Appl. Sci.* **2023**, *13*, 8728. <https://doi.org/10.3390/app13158728>.
3. Ali, I.; Ahmedy, I.; Gani, A.; Munir, M.U.; Anisi, M.H. Data Collection in Studies on Internet of Things (IoT), Wireless Sensor Networks (WSNs), and Sensor Cloud (SC): Similarities and Differences. *IEEE Access* **2022**, *10*, 33909–33931. <https://doi.org/10.1109/ACCESS.2022.3161929>.
4. Yang, Y.; Yang, T.; Wu, X.; Hu, B. UAV-Assisted Multi-Task Federated Learning with Task Knowledge Sharing. In *Proceedings of the ICC 2025—IEEE International Conference on Communications, Montreal, QC, Canada, 8–12 June 2025*; IEEE: New York, NY, USA, 2025; pp. 1548–1553. <https://doi.org/10.1109/ICC52391.2025.11162109>.
5. Li, K.; Ni, W.; Emami, Y.; Dressler, F. Data-Driven Flight Control of Internet-of-Drones for Sensor Data Aggregation Using Multi-Agent Deep Reinforcement Learning. *IEEE Wirel. Commun.* **2022**, *29*, 18–23. <https://doi.org/10.1109/MWC.002.2100681>.
6. Ladosz, P.; Weng, L.; Kim, M.; Oh, H. Exploration in deep reinforcement learning: A survey. *Inf. Fusion* **2022**, *85*, 1–22. <https://doi.org/10.1016/j.inffus.2022.03.003>.
7. Ekechi, C.C.; Elfouly, T.; Alouani, A.; Khattab, T. A Survey on UAV Control with Multi-Agent Reinforcement Learning. *Drones* **2025**, *9*, 484. <https://doi.org/10.3390/drones9070484>.

8. Ren, Z.; Rathinam, S.; Choset, H. A Conflict-Based Search Framework for Multiobjective Multiagent Path Finding. *IEEE Trans. Autom. Sci. Eng.* **2023**, *20*, 1262–1274. <https://doi.org/10.1109/TASE.2022.3183183>.
9. Chan, S.H.; Stern, R.; Felner, A.; Koenig, S. Greedy Priority-Based Search for Suboptimal Multi-Agent Path Finding. *Proc. Int. Symp. Comb. Search* **2023**, *16*, 11–19. <https://doi.org/10.1609/socs.v16i1.27278>.
10. Liang, P.; Li, Z.; You, D.; Li, Z. A Conflict-free multi-agent path planning method based on Improve A-star algorithm. In *Proceedings of the 2024 4th International Conference on Electrical Engineering and Control Science (IC2ECS), Nanjing, China, 27–29 December 2024*; IEEE: New York, NY, USA, 2024; pp. 799–806. <https://doi.org/10.1109/IC2ECS64405.2024.10928312>.
11. Wang, S.; Xu, H.; Zhang, Y.; Lin, J.; Lu, C.; Wang, X.; Li, W. Where Paths Collide: A Comprehensive Survey of Classic and Learning-Based Multi-Agent Pathfinding. *arXiv* **2025**, arXiv:2505.19219. <https://doi.org/10.48550/arXiv.2505.19219>.
12. Bogrybayeva, A.; Meraliyev, M.; Mustakhov, T.; Dauletbayev, B. Machine Learning to Solve Vehicle Routing Problems: A Survey. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 4754–4772. <https://doi.org/10.1109/TITS.2023.3334976>.
13. Cheikhrouhou, O.; Khoufi, I. A comprehensive survey on the Multiple Traveling Salesman Problem: Applications, approaches and taxonomy. *Comput. Sci. Rev.* **2021**, *40*, 100369. <https://doi.org/10.1016/j.cosrev.2021.100369>.
14. Murray, C.C.; Chu, A.G. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transp. Res. Part C Emerg. Technol.* **2015**, *54*, 86–109. <https://doi.org/10.1016/j.trc.2015.03.005>.
15. Wang, Z.; Sheu, J.B. Vehicle routing problem with drones. *Transp. Res. Part B Methodol.* **2019**, *122*, 350–364. <https://doi.org/10.1016/j.trb.2019.03.005>.
16. Koenig, S.; Likhachev, M. Fast Replanning for Navigation in Unknown Terrain. *IEEE Trans. Robot.* **2005**, *21*, 354–363. <https://doi.org/10.1109/TRO.2004.838026>.
17. Koenig, S.; Likhachev, M.; Furcy, D. Lifelong Planning A*. *Artif. Intell.* **2004**, *155*, 93–146. <https://doi.org/10.1016/j.artint.2003.12.001>.
18. Lee, S.; Joo, H.; Kim, K.; Kim, H. Rolling-Horizon Genetic Algorithm for Adaptive Path Planning in Hazardous Environments. *Comput. Electr. Eng.* **2026**, *129*, 110820. <https://doi.org/10.1016/j.compeleceng.2025.110820>.
19. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347. <https://doi.org/10.48550/arXiv.1707.06347>.
20. Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; Mordatch, I. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. *arXiv* **2017**, arXiv:1706.02275. <https://doi.org/10.48550/arXiv.1706.02275>.
21. Yu, C.; Velu, A.; Vinitzky, E.; Gao, J.; Wang, Y.; Bayen, A.; Wu, Y. The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games. *arXiv* **2022**, arXiv:2103.01955. <https://doi.org/10.48550/arXiv.2103.01955>.
22. Su, K.; Qian, F. Multi-UAV Cooperative Searching and Tracking for Moving Targets Based on Multi-Agent Reinforcement Learning. *Appl. Sci.* **2023**, *13*, 11905. <https://doi.org/10.3390/app132111905>.
23. Liu, Y.; Li, X.; Wang, J.; Wei, F.; Yang, J. Reinforcement-Learning-Based Multi-UAV Cooperative Search for Moving Targets in 3D Scenarios. *Drones* **2024**, *8*, 378. <https://doi.org/10.3390/drones8080378>.
24. Liao, W.C.; Wu, T.R.; Wu, I.C. Dynamic Sight Range Selection in Multi-Agent Reinforcement Learning. *arXiv* **2025**, arXiv:2505.12811. <https://doi.org/10.48550/arXiv.2505.12811>.
25. Sahingoz, O.K. Mobile networking with UAVs: Opportunities and challenges. In *Proceedings of the 2013 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 28–31 May 2013*; IEEE: New York, NY, USA, 2013; pp. 933–941. <https://doi.org/10.1109/ICUAS.2013.6564779>.
26. Gu, D.; Pei, G.; Ly, H.; Gerla, M.; Hong, X. Hierarchical routing for multi-layer ad-hoc wireless networks with UAVs. In *Proceedings of the MILCOM 2000 Proceedings, 21st Century Military Communications, Architectures and Technologies for Information Superiority (Cat. No.00CH37155), Los Angeles, CA, USA, 22–25 October 2000*; IEEE: New York, NY, USA, 2000; Volume 1, pp. 310–314. <https://doi.org/10.1109/MILCOM.2000.904964>.
27. Naous, T.; Itani, M.; Awad, M.; Sharafeddine, S. Reinforcement Learning in the Sky: A Survey on Enabling Intelligence in NTN-Based Communications. *IEEE Access* **2023**, *11*, 19941–19968. <https://doi.org/10.1109/ACCESS.2023.3236801>.
28. Umer, M.; Mohsin, M.A.; Nasir, A.A.; Abou-Zeid, H.; Hassan, S.A. Intelligent Spectrum Sharing in Integrated TN-NTNs: A Hierarchical Deep Reinforcement Learning Approach. *IEEE Wirel. Commun.* **2025**, *32*, 64–71. <https://doi.org/10.1109/MWC.001.2400421>.
29. Xin, Y.; Li, L.; Ning, Y.; Yang, Y.; Shi, P. UAV-Assisted Cooperative Charging and Data Collection Strategy for Heterogeneous Wireless Sensor Networks. *Drones* **2025**, *9*, 859. <https://doi.org/10.3390/drones9120859>.
30. Dechicha, M.I.; Eudes, A.; De Nazelle, P.; Marzat, J. Exploration Strategies for a Heterogeneous UAV-UGV Tandem. In *Proceedings of the 2025 European Conference on Mobile Robots (ECMR), Padova, Italy, 2–5 September 2025*; IEEE: New York, NY, USA, 2025; pp. 1–8. <https://doi.org/10.1109/ECMR65884.2025.11163236>.
31. Lu, F.; Mankoe, M.; Bi, H. A regulatory-no-fly-zone-aware drone-rider takeouts routing optimization. *Comput. Ind. Eng.* **2026**, *216*, 111992. <https://doi.org/10.1016/j.cie.2026.111992>.

32. Tang, J.; Duan, X.; Zhou, J.; Qu, K.; Ho, I.W.H.; Tian, D. Energy-Efficient Data Collection and Task Offloading Optimization in Heterogeneous Multi-Tier AAV Systems via Deep Reinforcement Learning. *IEEE Trans. Veh. Technol.* **2026**, *75*, 6732–6745. <https://doi.org/10.1109/TVT.2025.3622980>.
33. Le, C.; Vu, T.X.; Chatzinotas, S. Cooperative UAVs with Asynchronous Multi-agent Learning for Remote Data Collection. In *Proceedings of the 2024 IEEE Globecom Workshops (GC Wkshps), Cape Town, South Africa, 8–12 December 2024*; IEEE: New York, NY, USA, 2024; pp. 1–6. <https://doi.org/10.1109/GCWkshp64532.2024.11101581>.
34. Lee, C.; Lee, S.; Kim, T.; Bang, I.; Lee, J.H.; Chae, S.H. Multi-Agent Deep Reinforcement Learning-Based Multi-UAV Path Planning for Wireless Data Collection and Energy Transfer. In *Proceedings of the 2024 Fifteenth International Conference on Ubiquitous and Future Networks (ICUFN), Budapest, Hungary, 2–5 July 2024*; IEEE: New York, NY, USA, 2024; pp. 500–504. <https://doi.org/10.1109/ICUFN61752.2024.10625275>.
35. Zhang, Y.; Zhao, Q.; Mao, P.; Bai, Q.; Li, F.; Pavlova, S. Design and Control of an Ultra-Low-Cost Logistic Delivery Fixed-Wing UAV. *Appl. Sci.* **2024**, *14*, 4358. <https://doi.org/10.3390/app14114358>.
36. Halder, S.; Afsari, K. Robots in Inspection and Monitoring of Buildings and Infrastructure: A Systematic Review. *Appl. Sci.* **2023**, *13*, 2304. <https://doi.org/10.3390/app13042304>.
37. Khanzada, H.R.; Maqsood, A.; Basit, A. Reinforcement learning for UAV flight controls: Evaluating continuous space reinforcement learning algorithms for fixed-wing UAVs. *PLoS ONE* **2025**, *20*, e0334219. <https://doi.org/10.1371/journal.pone.0334219>.
38. Richter, D.J.; Calix, R.A.; Kim, K. A Review of Reinforcement Learning for Fixed-Wing Aircraft Control Tasks. *IEEE Access* **2024**, *12*, 103026–103048. <https://doi.org/10.1109/ACCESS.2024.3433540>.
39. Hu, T.; Zhu, W.; Yan, Y. Reward Hacking in Reinforcement Learning and RLHF: A Multidisciplinary Examination of Vulnerabilities, Mitigation Strategies, and Alignment Challenges. In *Proceedings of the 2025 5th Intelligent Cybersecurity Conference (ICSC), Tampa, FL, USA, 19–22 May 2025*; IEEE: New York, NY, USA, 2025; pp. 272–275. <https://doi.org/10.1109/ICSC65596.2025.11140447>.
40. Fu, J.; Zhao, X.; Yao, C.; Wang, H.; Han, Q.; Xiao, Y. Reward Shaping to Mitigate Reward Hacking in RLHF. *arXiv* **2025**, arXiv:2502.18770. <https://doi.org/10.48550/arXiv.2502.18770>.
41. Bai, Z.; Wu, G.; Barth, M.J.; Liu, Y.; Akin Sisbot, E.; Oguchi, K.; Huang, Z. A Survey and Framework of Cooperative Perception: From Heterogeneous Singleton to Hierarchical Cooperation. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 15191–15209. <https://doi.org/10.1109/TITS.2024.3436012>.
42. Furnon, V.; Perron, L. *OR-Tools Routing Library*, Version 9.14; Google: Mountain View, CA, USA, 2025.
43. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. <https://doi.org/10.1038/nature14236>.
44. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. A Brief Survey of Deep Reinforcement Learning. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. <https://doi.org/10.1109/MSP.2017.2743240>.
45. Konda, V.; Tsitsiklis, J. Actor-Critic Algorithms. In *Proceedings of the Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 1999; Volume 12.
46. Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; Abbeel, P. High-Dimensional Continuous Control Using Generalized Advantage Estimation. *arXiv* **2018**, arXiv:1506.02438. <https://doi.org/10.48550/arXiv.1506.02438>.
47. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
48. Lai, J.; Wang, D.; Zhao, Y. IMSBA: A Novel Integrated Sensing and Communication Beam Allocation Based on Multi-Agent Reinforcement Learning for mmWave Internet of Vehicles. *Appl. Sci.* **2025**, *15*, 6069. <https://doi.org/10.3390/app15116069>.
49. Kang, C.; Xu, J.; Bian, Y.; Tian, W. Path Navigation for Multi-Agent Flight System Based on Distributed Model Predictive Control and Reinforcement Learning. *Appl. Sci.* **2025**, *15*, 2234. <https://doi.org/10.3390/app15042234>.
50. Zhang, G.; Li, X.; Hu, G.; Li, Y.; Wang, X.; Zhang, Z. MARL-Based Multi-Satellite Intelligent Task Planning Method. *IEEE Access* **2023**, *11*, 135517–135528. <https://doi.org/10.1109/ACCESS.2023.3337358>.
51. Bettini, M.; Kortvelesy, R.; Blumenkamp, J.; Prorok, A. VMAS: A Vectorized Multi-agent Simulator for Collective Robot Learning. In *Distributed Autonomous Robotic Systems*; Springer Proceedings in Advanced Robotics; Bourgeois, J., Paik, J., Piranda, B., Werfel, J., Hauert, S., Pierson, A., Hamann, H., Lam, T.L., Matsuno, F., Mehr, N., et al., Eds.; Springer Nature: Cham, Switzerland, 2024; Volume 28, pp. 42–56. https://doi.org/10.1007/978-3-031-51497-5_4.

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.