remote sensing

MDPI

Article

Emergency-Oriented Multi-Satellite SAR Incremental Replanning Using Multi-Relational Graph Convolution and Constrained Proximal Policy Optimization

Xinying Wu [1], Chengzhi Sun [1,2,*] and Lingyu Bi [1]

[1] School of Remote Sensing and Geomatics Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China; 202311640006@nuist.edu.cn (X.W.); 202511770001@nuist.edu.cn (L.B.)
[2] Engineering Innovation Center for Integrated Remote Sensing and Navigation Applications, Ministry of Natural Resources, Nanjing 210044, China
* Correspondence: 003453@nuist.edu.cn

Highlights

What are the main findings?

- Across both equal-budget and natural-termination evaluations, GCN-PPO consistently produced feasible schedules without hard-constraint violations while maintaining high deadline compliance and causing less disruption to the original plan than the learning and iterative-search baselines. This stability advantage persisted across the held-out test set and became more pronounced as candidate competition increased.

What are the implications of the main findings?

- By replacing request-specific iterative search with offline relational learning and online constrained inference, the method supports SAR emergency replanning under an explicit response deadline. Its incremental-editing mechanism limits changes to local conflicts involving emergency tasks, thereby protecting unaffected parts of the original plan in high-load scenarios with large candidate sets.

Abstract

Floods, earthquakes, and landslides may require a multi-satellite synthetic aperture radar (SAR) system to insert new observations after a regular plan has been prepared but before upload. The revised plan must respect visibility, imaging chronology, look-direction switching, orbit resources, strip formation, and plan stability. We propose a multi-relational graph convolutional network with constrained proximal policy optimization (GCN-PPO). Five opportunity relations encode conflict, feasible sequence, cross-satellite competition, demand alternatives, and strip synergy. A three-level policy selects the action category, logical completion unit, and observation opportunity, while feasibility-normalized constraints regulate emergency-service shortfall and plan perturbation. In a 300 s equal-budget experiment, all 750 returned schedules were feasible and free of hard-constraint violations, and 96.4% passed the preregistered timing contract. Under natural termination, GCN-PPO responded in 102.4 s on average, 9.35% of the ALNS time, and met the normalized emergency target in 98.1% of evaluations. Its per-edit perturbation was 31.4% lower than MLP-PPO and 35.9% lower than ALNS. Timely executable schedules support rapid SAR data acquisition for downstream disaster-object detection and emergency monitoring.

Check for updates

Academic Editor: Guangcai Sun

Received: 28 August 2026
Revised: 16 October 2026
Accepted: 20 October 2026
Published: 26 October 2026

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

Keywords: multi-satellite mission planning; synthetic aperture radar; emergency remote sensing; reinforcement learning; dynamic replanning

1. Introduction

1.1. Background

Earth observation systems are evolving from a small number of high-value satellites toward multi-orbit, networked constellations with short revisit intervals. Remote sensing data support disaster monitoring, land surveys, crop-yield estimation, and water-environment management, and constellation operations place greater demands on unified resource management, rapid conflict resolution, and emergency response [1,2]. In a centralized ground-planning workflow, operators specify the planning horizon, participating satellites, task locations, priorities, and required observation counts. The ground system then receives orbit predictions or precomputed visibility results and generates a task sequence that satisfies payload and platform constraints. The scheduler therefore handles a request-defined combination of variable resources, demands, and observation opportunities rather than an abstract ranking problem of fixed size.

Floods, earthquakes, landslides, typhoons, and reservoir emergencies can change observation priorities within minutes. Emergency managers need to decide which affected areas should be observed first and whether upcoming satellite passes can provide a usable first image. SAR is independent of solar illumination and remains effective through cloud, rain, smoke, and haze, which makes it valuable for flood mapping, landslide and dam monitoring, surface-deformation measurement, and rapid damage assessment. In this workflow, mission planning determines whether the first useful imaging opportunity is scheduled before it passes. Here, timeliness means producing a revised imaging schedule after an emergency request arrives and before plan upload; data downlink and product generation are outside the scope of this study.

Multi-satellite remote sensing scheduling is oversubscribed, combinatorial, and strongly coupled in time [3–5]. For a SAR payload, imageability depends on the overflight, incidence angle, left/right look direction, continuous operation, beam-command segments, and orbit-level resources. A demand may have several alternatives across satellites and orbits. Opportunities on the same satellite can be mutually exclusive because of time overlap or can be combined into a continuous strip when their look direction and timing are compatible. Gridding an areal task further increases both the number of logical completion units and the number of candidate opportunities. A priority-only rule can consequently consume scarce windows too early.

Operations also require temporary emergency tasks to be inserted after the regular plan has been generated. At that point, the schedule has not yet been uploaded or executed, so pending tasks may be adjusted, but discarding the entire regular plan is undesirable. A practical dynamic planner should edit the existing plan incrementally: retain strips that do not block the last feasible emergency opportunities, remove only the conflicting segments that are necessary, and quantify the disruption caused by canceling high-priority, near-execution, or hard-to-replace tasks. This setting combines an emergency-service floor, a plan-stability requirement, and an online response deadline. A single fixed reward weight does not directly enforce all three requirements.

1.2. Related Work

1.2.1. Exact Optimization, Rules, and Metaheuristics

Early work on Earth observation planning used time-window discretization, mathematical programming, constraint propagation, and constructive heuristics. Wolfe and Sorensen compared three scheduling algorithms for Earth-observing systems [6]. Lemaître et al. studied observation selection and sequencing for agile satellites [7], and Bianchessi and Righini developed planning algorithms for the COSMO-SkyMed constellation [8]. Later studies represented profit and maneuver time as a time-dependent orienteering problem [9] and introduced exact, divide-and-conquer, and parallel-machine formulations to improve interpretability and optimality guarantees [10–12]. These methods are attractive for small instances and stringent audit requirements, but their solution time rises quickly with the number of candidates.

Genetic algorithms and adaptive large neighborhood search (ALNS) use randomized global search or destroy-and-repair iterations to obtain high-quality solutions within a practical time [13–16]. Dynamic neighborhood methods can insert and replace tasks around an existing schedule [17]. Li et al. formulated a multi-scenario mixed-integer model and combined priority screening, rolling horizons, and ALNS for multi-satellite planning [18]. These methods express constraints directly and require no offline training, so they are essential engineering baselines for this study. However, each new request generally requires repeated decoding and neighborhood evaluation, and performance depends on the neighborhood design, initial solution, and wall-clock budget.

1.2.2. Deep Reinforcement Learning and Graph Representation Learning

Deep reinforcement learning casts combinatorial scheduling as sequential decision making and shifts part of the search cost to offline training. Existing studies include DQN- or PPO-based satellite scheduling, parameter transfer, multi-agent cooperation, and generic Markov decision process formulations [19–24]. Graph neural networks can produce transferable representations from task adjacency. Chun et al. combined a time-attitude graph with PPO [25], while Wang et al. learned a multi-satellite policy on a heterogeneous graph [26]. Attention-based decision networks and two-stage hybrid optimization have further improved construction quality [27,28]. Dynamic graph reinforcement learning, sequence-based replanning, and variable-duration imaging demonstrate the potential of learned policies for event response [29–31]. Hybrid graph-metaheuristic methods and task denoising address large instances by reducing the candidate set [32–34]. Wang et al. [26] already represent heterogeneous interactions with edge-specific attention; we do not claim to introduce heterogeneous graph scheduling. The distinction examined here is the five SAR strip-opportunity relations and the constraint-aware editing of an existing plan after an emergency request.

Three gaps remain. First, many graph models emphasize adjacency or heterogeneous node types but do not explicitly separate edge semantics such as hard conflict, feasible sequence, alternatives, and synergy. Second, a fixed output layer does not naturally accommodate simultaneous changes in the numbers of satellites, logical demands, and observation opportunities. Third, dynamic planners often use fixed weights to trade off emergency service against perturbation or to reconstruct a schedule from scratch, making it difficult to maintain both emergency service and original-plan retention. A uniform absolute service threshold can also exceed the attainable level in an overloaded scenario, producing a training signal that is inconsistent with scenario feasibility.

1.3. Objectives and Contributions

This study addresses centralized ground planning for multiple SAR satellites. Emergency requests arrive as a batch after the regular plan is generated, and the revised schedule must be available before upload. We therefore evaluate executable delivery, emergency service, and plan stability within a deadline instead of treating unlimited-time scalar quality as the only endpoint. The main contributions are as follows:

- We formulate request-driven multi-satellite SAR planning in a common model that includes the selected satellites, a 24 h horizon, regular grids, emergency units, visibility opportunities, the original plan, and a shared hard-constraint audit.
- A five-relation opportunity graph and explicit scale-load state describe conflict, temporal sequence, cross-satellite competition, demand alternatives, and strip synergy, allowing one policy to process 9, 18, and 36 satellites.
- A three-level variable-length action policy and constrained PPO estimate planning return, emergency-service shortfall, and original-plan perturbation separately, with constraint pressure adjusted to a conservative scenario-specific schedulability reference.
- A protection set limits replanning to the local conflicts that affect emergency feasibility. Family-wise data splitting, five training seeds, the stage-limited test, and natural-termination runs examine deadline compliance, constraint satisfaction, and plan stability. Section 2.3.3 specifies the static stopping criteria used by the metaheuristic baselines.

Terminology is used consistently as follows: Mission planning denotes the complete process that transforms observation requests and orbital information into an executable multi-satellite plan. Task planning and scheduling denote the selection and temporal sequencing of observation actions under resource and physical constraints, whereas replanning refers specifically to incremental revision of an existing static schedule after emergency demands arrive. An opportunity is a raw satellite-target visibility window; a meta-task is a task-specific imaging instance materialized from one opportunity (compatible meta-tasks may be pre-aggregated into a candidate strip); and a logical completion unit is the demand-level entity whose required observations determine task completion.

2. Materials and Methods

2.1. Workflow and Mathematical Model

A planning request is denoted by $Q = (S, H, T^R, T^E, O, P^0)$. Here, S is the ground-operator-selected subset of available satellites; $H = [t_0, t_0 + 24\text{ h}]$ is the planning horizon; T^R and T^E are the regular and emergency logical demands, respectively; O is the set of computed visibility opportunities; and P^0 is the regular static plan. The emergency event occurs after P^0 is generated but before schedule upload and execution. A random seed controls network initialization, policy sampling, and stochastic optimization but does not alter the request definition.

Planning is centralized on the ground. Each request specifies the participating satellites and a continuous 24 h horizon. The regular schedule is prepared first, after which a batch of emergency demands arrives; each emergency demand requires one valid observation within the horizon. The simulations use fixed two-line element snapshots with Simplified General Perturbations 4 (SGP4) propagation; when visibility opportunities have already been computed, the same precomputed opportunity set is supplied directly. All algorithms therefore receive the same opportunity set.

Figure 1 illustrates the scheduling setting. Each valid satellite-target visibility opportunity becomes a meta-task, and candidate meta-tasks can conflict through shared time, maneuver, resource, or demand-completion conditions.

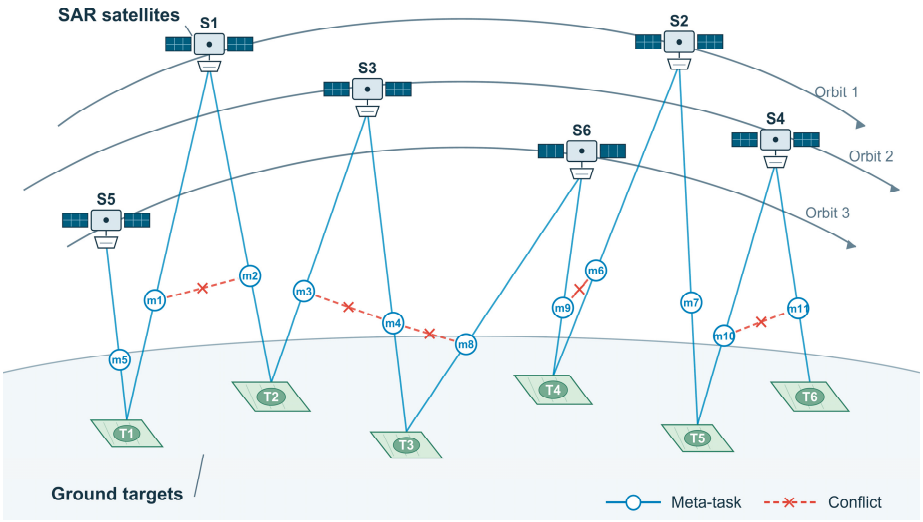


Figure 1. Multi-satellite, multi-orbit SAR planning scenario. Each satellite-target visibility opportunity forms a meta-task; dashed red links show representative conflicts between candidate meta-tasks.

For each satellite and orbit, the cumulative imaging time is limited to 240 s, and the number of beam-command segments is limited to 16. Nominal left/right platform-roll states are fixed at ±30°, and consecutive observations on opposite sides require at least 130 s. These limits serve as engineering proxies for energy and storage; thermal constraints are not modeled separately. Regular coverage counts only unique grid cells with at least one valid opportunity during the 24 h period, and emergency units pass the same visibility audit.

The complete planning path is shown in Figure 2. The static phase converts the mission request and orbital data into an audited regular schedule. After the emergency batch arrives, the planner updates affected opportunities, protects nonblocking elements of the original plan, and solves the remaining local edit problem with the same constrained policy.

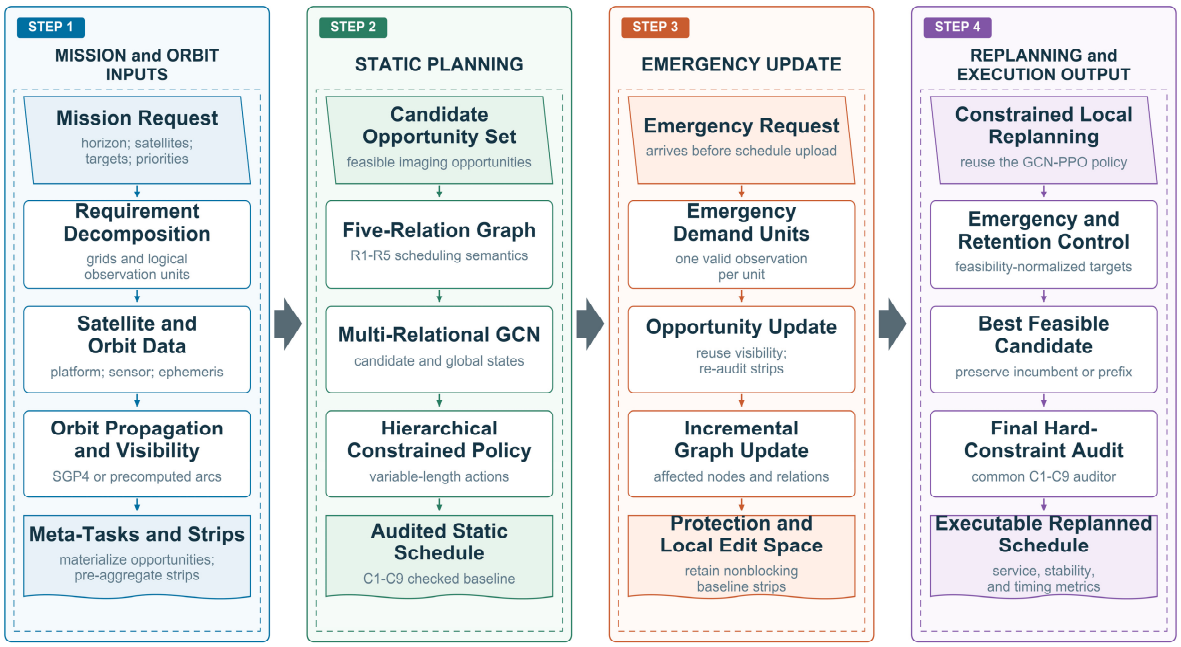


Figure 2. End-to-end workflow for regular static planning and emergency incremental replanning. The static schedule supplies the original-plan baseline, while updated visibility opportunities define the local edit space for emergency insertion.

2.1.1. Visibility, Meta-Tasks, and Strip Candidates

Regular areal targets are approximated by grids derived from provincial or municipal administrative boundaries. Emergency areal targets are divided into observation units according to the input range and azimuth swath widths. A meta-task o_i is generated for each satellite, target unit, and visibility arc. Its attributes include satellite, orbit, start and end times, zero-Doppler time, look direction, incidence angle, beam mode, logical demand, and covered units. The simulations propagate the orbit with SGP4 [35]. In operations, O can be replaced by visibility results from a precise ground orbit system without changing the downstream model.

The nominal 30-degree left/right setting specifies the assumed payload pointing state relative to nadir. The incidence angle is a separate opportunity-level geometric quantity measured between the line of sight and the local ground normal. It is checked independently against the 17-50-degree interval in C6; the nominal look setting does not replace this feasibility check.

A point target is placed at the center of a 20 km × 20 km scene. Let v_i denote satellite velocity and t_i^z the zero-Doppler time. The azimuth imaging duration d_i is 20 km/v_i, and the imaging interval is symmetric about t_i^z. This definition avoids a zero-second duration for point targets and derives the interval entirely from scene size, satellite velocity, and the visibility arc.

Equation (1) is a scheduling-level duration proxy for a fixed 20 km × 20 km scene centered at zero Doppler. The velocity term is a circular-orbit inertial-speed proxy computed from orbital altitude, not a measured ground-footprint speed. The proxy deliberately ignores incidence angle and left/right-look effects on ground speed; it supplies a common nonzero resource cost for every method and is not a SAR acquisition or focusing model.

$$d_i = \frac{20\ \text{km}}{v_i}, [s_i, e_i] = \left[t_i^z - \frac{d_i}{2}, t_i^z + \frac{d_i}{2} \right] \tag{1}$$

Meta-tasks are sorted by satellite and orbit before strip pre-aggregation. Opportunities on the same satellite and orbit that share a look direction and are temporally contiguous and spatially adjacent may be combined into a long strip. The merged strip retains every beam segment, start and end time, target key, and covered unit. Continuous imaging with one beam mode counts as one command segment; different beam modes are counted as separate segments. Preprocessing retains both the complete long strip and deterministic split alternatives, preserving strip synergy without losing useful local combinations. After an emergency arrives, regular and emergency opportunities are re-audited and may form a mixed strip when timing and beam constraints permit.

2.1.2. Sets, Variables, and Objective

Let S and R denote the participating satellites and orbits. Let K^R and K^E denote the regular and emergency completion units visible in the 24 h horizon. Let I denote the pre-aggregated and pruned strip set, and let $\Omega(k)$ denote the candidates that can complete unit k. The binary variable x_i indicates whether strip i is selected; d_i and b_i are its imaging duration and number of beam-command segments, while p_k and r_k denote priority and required observation count. For an original strip j, y_j records retention in the revised plan. C^R, C^E, and L^E denote regular coverage, emergency-unit satisfaction, and emergency logical-task completion; R^P and D^P denote weighted original-plan retention and perturbation cost.

The static phase maximizes regular coverage, task priority, and resource efficiency. The dynamic phase additionally rewards emergency service and original-plan retention. Equation (2) defines the normalized planning-quality score J. Response deadline and

perturbation stability are reported as separate engineering endpoints rather than being hidden inside one scalar objective.

$$J = 10\left(C^R + 6C^E + 3L^E + 0.25T^E + 0.35R^P - 0.02C^{time} - 0.01C^{cmd}\right) \quad (2)$$

In Equation (2), C^{time} and C^{cmd} denote the scenario-normalized imaging time and beam-command costs, respectively. The emergency-earliness term T^E and the overall factor of 10 follow the common evaluation procedure. Equation (2) describes soft objectives only. A shared planning state, action mask, and output audit strictly enforce physical executability; hard constraints are not approximated with finite penalties. In the dynamic phase, emergency-service shortfall and plan perturbation become constraint costs rather than larger fixed reward terms.

Equation (2) is the common final evaluation score J used in the result tables. The PPO action-level reward uses corresponding service, retention, and resource terms, but its stepwise returns and Lagrange-weighted costs are a training surrogate rather than an identical full-plan objective. The GA and the ALNS-style baseline rank decoded candidate sequences using the shared PlanningState action reward and C1-C9 feasibility auditor; the final J is recomputed by the common evaluator. Priority-EDF follows a fixed emergency-priority/deadline order and does not directly optimize J. The fixed-weight ablation holds both initial multipliers at 1.0; it is a mechanism ablation rather than a validation-optimized static-weight competitor. This comparison isolates multiplier adaptation at the stated reference weights.

2.1.3. Hard Constraints

The shared decoder enforces nine hard constraints. A candidate interval must lie within a valid visibility arc and the 24 h horizon (C1). Imaging intervals and required transitions on the same satellite must not overlap, so conflicting candidates satisfy $x_i + x_j \leq 1$ (C2). Cumulative imaging time and beam-command segments per orbit cannot exceed 240 s and 16, respectively (C3 and C4); continuous imaging with one beam mode counts as one command segment.

Consecutive observations on opposite sides require at least 130 s (C5), and the incidence angle must remain between 17 and 50 degrees (C6). A logical demand k is complete only if the number of selected candidates in $\Omega(k)$ reaches r_k (C7), and a completion unit cannot be rewarded more than r_k times (C8). During dynamic replanning, D^P must remain within its budget, and a strip in the protection set cannot be removed (C9). Invalid actions are excluded before policy sampling.

2.1.4. Feasibility-Normalized Dynamic Constraints

In an overloaded scenario, a fixed absolute emergency rate may exceed what the available resources can deliver. The materialization stage therefore groups candidates that satisfy C1-C7 by emergency completion unit and retains at most four representative opportunities per group, chosen for satellite, orbit, and time-window diversity. In priority and last-opportunity order, these representatives are inserted using the same PlanningState constraint decoder as formal scheduling, and an insertion is accepted only if the current partial plan remains feasible. The number of jointly serviceable units divided by the number of visible emergency units defines $q^E(Q)$. This quantity is a reproducible conservative schedulability reference, not a theoretical upper bound or the true optimal service rate. The current dataset does not separately calibrate a logical-task reference, so $q^L(Q) = q^E(Q)$; later datasets can estimate them independently.

$$C^E \geq \tau^E q^E(Q), L^E \geq \tau^L q^L(Q), R^P \geq \tau^P \quad (3)$$

The experiments use $\tau^E = 0.85$, $\tau^L = 0.80$, and $\tau^P = 0.45$, with a maximum perturbation cost equal to 0.60 of the original-plan size. The retention target $\tau^P = 0.45$ is a training-time constrained-RL reference rather than a hard per-instance feasibility requirement; shortfall relative to this reference updates λ_P, whereas C1-C9 remain the per-instance hard feasibility conditions. The emergency test asks whether the policy reaches the conservative schedulability reference for the current scenario; it does not claim that every emergency unit can be completed.

2.2. Multi-Relational GCN-PPO

2.2.1. Multi-Relational Opportunity Graph

Each candidate strip is represented by a node v_i. The model uses 13 node features: priority, normalized imaging duration, time of day, look direction, longitude, latitude, emergency indicator, incidence angle, remaining-day-time proxy, coverage output, original-plan indicator, hard-conflict degree, and cross-satellite competition intensity. The graph also stores four pair diagnostics (two continuous and two binary) and a five-dimensional relation encoding. Relation type determines the message channel. The pair diagnostics aid interpretation and consistency auditing; the discrete R1-R5 rules construct edges, and relation type selects the message channel. Five directed relations are defined from physical and operational semantics.

For exact reproducibility, the node vector is x = [p, d/240, t_day/86400, s, lon/180, lat/90, e, theta/50, m_day, y_cover, o_plan, c_conflict, c_competition]. Here, p is priority, d is the imaging duration, t_day is the start time within the 24 h horizon, s maps left/unknown/right to 0/0.5/1, e is the emergency flag, and theta is the incidence angle. The ninth component m_day is a clipped remaining-time-within-day proxy, not an actual task-deadline slack. Coverage yield uses log1p normalization by the request maximum; o_plan marks the original plan, and the last two components are normalized structural-count diagnostics. The four pair diagnostics are r_time (interval-overlap ratio) and r_space (one minus a clipped planar longitude/latitude radian-distance proxy divided by pi), both continuous, plus b_side (look-side switch) and b_sat (same-satellite flag), both binary. These values aid interpretation and consistency audits; the discrete R1-R5 rules construct edges, and relation type selects the RGCN message channel. Appendix A Note A2 gives the exact formulas for the remaining-day proxy and the two structural-count features.

R1 represents same-satellite hard conflict caused by overlapping imaging intervals or an infeasible look-direction transition; orbit resource capacities are enforced by the shared planning state rather than an R1 edge. R2 links executable temporal sequences on one satellite. R3 connects opportunities on different satellites that serve nearby demands. R4 links alternative observations of a shared logical completion unit. R5 connects continuous same-orbit opportunities whose validated metadata permit merging and resource savings.

The deterministic relation rules are as follows: R1 is bidirectional for same-satellite interval overlap or an opposite-look transition gap below 130 s; positive-duration intervals are half-open, and coincident point events conflict. R2 is a directed earlier-to-later edge for same-satellite, same-orbit opportunities when the transition is feasible and the earlier-start to later-end span is at most 240 s. R3 is bidirectional for different satellites with great-circle separation strictly below 3 degrees and start-time difference strictly below 30 min; nearby demand means candidate target locations within this angular threshold. R4 is bidirectional for opportunities sharing a logical completion key. R5 is bidirectional for same-satellite, same-orbit, same-side materialized strips with disjoint logical keys, valid merged metadata, a span at most 240 s, and a third candidate containing both metadata sets. Pairs use (start time, end time, and original index) ordering; relation tuples are lexicographically sorted,

and parallel relation types are retained. Appendix A Table A1 summarizes the deterministic construction rules for all five relations.

Figure 3 gives a compact example of the graph supplied to the encoder. In the label q_i/O_j/L (or R), q_i is a schematic single logical completion unit, O_j identifies the orbit/pass, and L/R is the left/right look direction. A merged materialized strip may serve a set $Q(v)$ of several units; the q_i label in this illustration does not restrict every strip to one demand. R4 follows shared members of $Q(v)$, while R2 and R5 also use orbit and strip lineage.

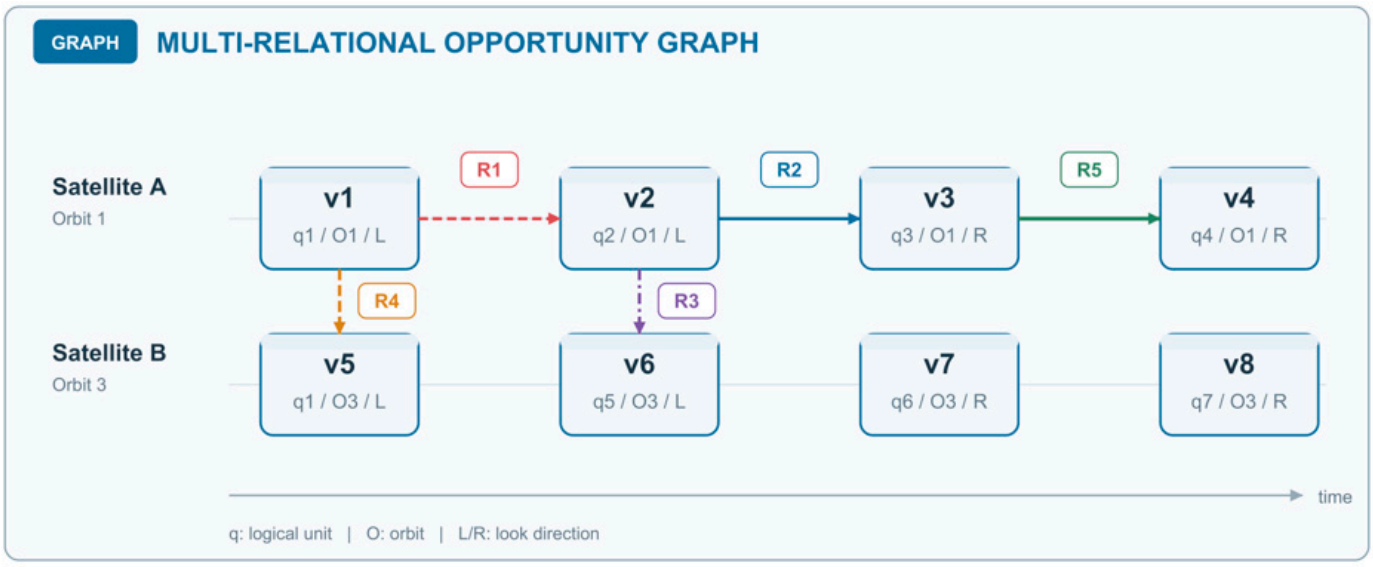


Figure 3. Multi-relational opportunity graph. R1-R5 denote hard conflict, feasible sequence, cross-satellite competition, demand alternatives and strip synergy. Labels q_i/O_j/L or R show a single-unit example; a merged strip may serve multiple units $Q(v)$.

2.2.2. Relation-Specific Graph Convolutional Encoder

The encoder contains three relation-specific graph convolutional layers and follows standard graph convolution and relational graph convolution formulations. Sparse message passing uses the PyTorch Geometric relational graph convolution (RGCNConv) operator [36–38]. Layer dimensions are 13 → 128 → 128 → 32, with zero dropout. Each layer applies an exponential linear unit, a linear residual projection, and layer normalization. Maximum aggregation retains the strongest neighbor signal within each sparse conflict or competition relation, after which the relation-channel outputs are summed.

$$h_i^{(l+1)} = \mathrm{LN}\left\{\mathrm{ELU}\left[W_0^{(l)} h_i^{(l)} + \sum\nolimits_r \max_{j \in N_r(i)} \left(W_r^{(l)} h_j^{(l)}\right)\right] + R^{(l)} h_i^{(l)}\right\} \quad (4)$$

Relation parameters W_r allow the encoder to distinguish conflict from synergy instead of mixing opposing meanings in a homogeneous adjacency matrix. Global mean, maximum, and standard deviation pooling are applied to the 32-dimensional node representations. Their concatenation is projected through a linear layer and hyperbolic tangent to obtain a 32-dimensional graph representation g. For a large graph, destination-oriented message passing processes core-node batches of at most 2048 nodes while retaining every incoming edge to the current batch.

Every constructed graph is audited before use. Edge counts for all five relations must agree with scenario semantics, and the continuous-strip relation cannot be overwritten by another relation. Bidirectionality, temporal orientation, and node indices are checked against the candidate strips. This structural audit is required before attributing an experimental effect to relation-aware message passing.

2.2.3. Unified State and Scale-Load Features

The planning state has 49 dimensions: a 32-dimensional graph representation, seven resource features, and ten scale-load features. The resource features describe the candidate satellite, remaining orbit imaging fraction, remaining command fraction, current look state, readiness for opposite-side switching, occupied-task fraction, and valid-action fraction. Scale-load features make resource and demand intensity explicit so that one model can operate across several constellation sizes.

2.2.4. Hierarchical Variable-Length Action Policy

Normalizing directly over tens of thousands of opportunities can prune valuable regular tasks too early and makes policy entropy incomparable between small and large instances. We factor the action a into action category c, logical completion unit k, and concrete opportunity i. The first level distinguishes emergency insertion or replacement, restoration of an original-plan opportunity, and non-disruptive regular addition. The second selects a logical completion unit. The third selects a concrete satellite orbit time beam opportunity from $\Omega(k)$. Protection and edit feasibility are enforced by the shared action mask and decoder rather than represented as separate learned action categories. The joint probability is

$$\pi_\theta(a|s) = \pi_\theta(c|s)\cdot\pi_\theta(k|c,s)\cdot\pi_\theta(i|k,c,s) \tag{5}$$

Each level is normalized under the shared hard-constraint mask. Candidate-pruning scores combine priority, opportunity scarcity, conflict degree, continuous-strip benefit, resource margin, and deadline margin. The logical-demand level always retains a representative for each unsatisfied demand, preventing global score truncation from removing a demand altogether. The normalized policy entropy is $\overline{\mathcal{H}} = \frac{\mathcal{H}}{\log|A_{\text{valid}}|}$, where $\mathcal{H}$ denotes the raw policy entropy, and A_{valid} is the set of currently valid actions under the mask. This normalization keeps exploration pressure comparable between small 9-satellite graphs and large 36-satellite graphs.

2.2.5. Constrained Proximal Policy Optimization

Proximal policy optimization limits update the magnitude by clipping the ratio between new and old action probabilities [39]. Generalized advantage estimation balances bias and variance in value estimates [40], and schedule construction follows a Markov decision process [41]. Separate value heads estimate return, emergency shortfall, and plan perturbation, avoiding the mixing of different scales in one value function. The clipped return objective is

$$L_{\text{clip}}(\theta) = E_t\Big(\min\Big(r_t(\theta)\widehat{A}_t, \text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\widehat{A}_t\Big)\Big) \tag{6}$$

Here, $r_t(\theta)$ is the policy-probability ratio, $\widehat{A}_t$ is the generalized advantage estimate, and ε is the clipping coefficient. Let $J^E(\theta)$ and $J^P(\theta)$ denote the expected discounted costs of emergency-service shortfall and original-plan disruption, respectively. Emergency shortfall is determined by the larger normalized deficit relative to $\tau^E q^E(Q)$ and $\tau^L q^L(Q)$, whereas the disruption cost accumulates stepwise decreases in R^P. The admissible cost thresholds $d^E(Q)$ and d^P convert the lower-bound service and retention conditions in Equation (3) into upper-bound cost constraints. The Lagrangian objective is

$$\max_\theta L_{\text{clip}}(\theta) - \lambda_E\big(J^E(\theta) - d^E(Q)\big) - \lambda_P\big(J^P(\theta) - d^P\big) \tag{7}$$

$$\lambda_j \leftarrow \text{clip}\Big(\lambda_j + \eta_j\big(J_j(\theta) - d_j\big), \lambda_{j,\min}, \lambda_{j,\max}\Big), j \in \{E, P\} \tag{8}$$

Here, η_j is the multiplier-specific learning rate, and $\lambda_{j,\min}$ and $\lambda_{j,\max}$ are the corresponding bounds. In the reported experiments, $\lambda_{E,\min} = 1.0$, $\lambda_{P,\min} = 0$, and $\lambda_{E,\max} = \lambda_{P,\max} = 20$.

The emergency multiplier is therefore clipped to at least its initial value so that emergency-constraint pressure cannot vanish, whereas the retention multiplier retains a configurable lower bound. Each multiplier increases when its constraint is violated and may decrease when the signed update indicates recovery; projection enforces its prescribed lower bound. A target Kullback-Leibler divergence terminates an update epoch early, and errors and return scales are logged separately for all three value heads.

2.2.6. Reward, Perturbation, and Scarce-Opportunity Protection

The step reward credits newly delivered service and applies small penalties for resource use. Regular coverage, emergency units, emergency logical tasks, imaging earliness, and original-plan retention are normalized by scenario size and weighted by 1, 6, 3, 0.25, and 0.35, respectively. Imaging time and beam-command costs are weighted by 0.02 and 0.01. Actions that violate a hard constraint are removed by the mask before sampling rather than discouraged by a finite negative reward.

For action i, ΔC_i^R, ΔC_i^E, and ΔL_i^E denote the incremental normalized gains in regular coverage, emergency-unit satisfaction, and emergency logical-task completion, respectively. E_i is the normalized earliness credit of a newly served emergency unit, and ΔR_i^P is the net normalized contribution to weighted original-plan retention after accounting for any original strips displaced by the action. The leading factor 10 is the global reward-scale constant. The terms $\frac{d_i}{D_time}$ and $\frac{b_i}{D_cmd}$ are the imaging time and beam-command costs normalized by request-level capacities. Here, D_time = max(240 s times the number of available satellite orbit pairs, 1 s), and D_cmd = max(16 times that number, 1).

$$r_i = 10\left[\Delta C_i^R + 6\Delta C_i^E + 3\Delta L_i^E + 0.25E_i + 0.35\Delta R_i^P - \frac{0.02d_i}{D_time} - \frac{0.01b_i}{D_cmd}\right] \tag{9}$$

Let $D^P = \sum_j D_j$ be the weighted perturbation cost of all canceled or split original-plan strips. The cost of one strip combines priority, time to execution, and the remaining number of alternatives:

$$D_j = \left[\frac{\sum_{k\in K_j} p_k}{n_j^{\text{alt}} + 1}\right] \cdot \left[1 + 0.5 \cdot \max\left(0, 1 - \frac{\Delta \text{t}_j}{6\ \text{h}}\right)\right] \tag{10}$$

$$d^{\text{edit}} = \frac{D^P}{\max(1, N_\Delta)}, d^{\text{size}} = \frac{D^P}{\max(1, N^0)} \tag{11}$$

Here, N_Δ is the number of canceled or split original strips, and N^0 is the number of pending original strips before the emergency arrival. The per-edit measure d^{edit} gives the mean cost of each necessary edit, whereas d^{size} measures disruption relative to the size of the original plan.

The mask does not exclude all regular actions merely because emergency demand is present. For each unsatisfied emergency unit, the planner estimates the last feasible opportunity and the remaining alternatives. A regular action is blocked only if it consumes the last opportunity, reduces joint schedulability below the constraint, or makes perturbation-budget recovery impossible. This risk-based rule allows valuable regular tasks to continue when emergency feasibility is preserved.

2.2.7. Static Planning and Incremental Editing

The static phase starts from an empty plan and selects logical demands and observation opportunities sequentially. The dynamic phase starts from P^0, marks its strips as retention candidates, and re-audits candidates related to emergency opportunities. Original strips that do not affect emergency feasibility enter the protection set; only the remaining portion can be edited. Here, incremental refers to local decisions that preserve protected elements

of the original plan. It does not assume that one graph maintenance implementation is inherently faster.

After an emergency event, the planner restores the original plan and updates affected candidates and relations. It estimates each emergency unit's last feasible opportunity, remaining alternatives, and joint schedulability reference and then protects original strips that do not reduce emergency feasibility. Conflicting original strips, open regular opportunities, and emergency opportunities are re-aggregated, and the hierarchical policy acts only in this local edit space. Every step checks hard constraints, emergency service, and the perturbation budget. The final plan is audited again against C1-C9 and is reported with retention, perturbation, and response time.

2.2.8. Offline Training and Online Inference

Offline training balances transition counts across satellite scale, actual load, and static/dynamic phase. A trajectory contains at most 512 steps. A truncated trajectory is bootstrapped from its value estimate, whereas a genuinely terminal state has zero bootstrap value. Each model is trained for 784 episodes and evaluated every 56 episodes on 50 validation requests. Five seeds are trained independently from random initialization, and training and validation sets are defined before evaluation on the held-out test set.

At inference, the model encodes the current candidate graph and scores hierarchical actions in batches without gradient updates. The emergency phase supports reproducible local graph maintenance and full-graph reconstruction; the latter is used only as an implementation diagnostic. Preprocessing, static planning, dynamic response, and end-to-end time are recorded separately. All timing results reported here were measured without a GPU and with one CPU thread per process.

2.3. Experimental Setup

2.3.1. Research Questions

The experiments first verify that all methods receive the same request, candidate strips, initial plan, and hard-constraint auditor; an infeasible schedule is counted as a failure. We then examine the engineering deadline, normalized emergency constraint, plan stability, and scale-dependent advantage region. Normalized planning quality is treated as a quality-control endpoint rather than the sole ranking criterion.

The evaluation asks whether GCN-PPO can deliver an executable post-arrival schedule within the nominal 300 s response allowance, meet the feasibility-normalized emergency constraint while controlling perturbation, separate the effects of relational representation and adaptive constraints, and retain its advantage as satellite count, load, candidate count, and conflict density increase. The stage-limited and natural-termination experiments characterize quality, response time, and plan stability under the submitted configurations.

2.3.2. Satellites, Tasks, and Scenario Dataset

The experiment begins with orbital parameters of operational SAR satellites and expands them to a simulated 36-satellite Walker-type constellation with nine orbital planes and four approximately equally phased satellites per plane. Each request selects 9, 18, or 36 participating satellites. The 9-satellite setting selects one satellite from each plane, the 18-satellite setting balances selection across all planes, and the 36-satellite setting uses the complete constellation. Satellites are numbered internally from 1 to 36; the identifiers do not encode performance rank.

The dataset contains 500 requests organized into 250 paired scenario families. The two members of a family share task locations, the satellite skeleton, and the planning horizon and represent proportional and high load. Entire families are assigned to 300 training,

50 validation, and 150 held-out test requests, corresponding to 150, 25, and 75 families. The validation and test subsets contain 150 and 300 emergency-task identifiers absent from training. This holdout prevents memorization of fixed task identifiers but is not a geographic block holdout, so the RQ4 generalization claim is limited to new task combinations. Table 1 summarizes the dataset split. The fixed scenario inputs and independent integrity validator are provided in Supplementary File S1.

Table 1. Scenario dataset design.

Dimension	Setting	Description
Planning horizon	24 h	A 24 h schedule is prepared one day before execution
Satellite scale	9/18/36	Satellites are balanced across orbital planes
Scenario families	250 families; 500 requests	Each family contains proportional- and high-load members
Split	300/50/150	Training/validation/unseen test; whole-family assignment
Actual load	Low/medium/high	Assigned after materialization using demand-capacity ratio, conflict density, and joint schedulability
Dynamic event	One batch arrival	Emergency demands are inserted as one batch after the regular plan; each unit requires one observation
Feasibility	Single-task visibility + joint schedulability	Emergency units without opportunities are removed; regular denominators retain only visible grids

2.3.3. Comparative Methods

Priority-earliest-deadline-first (Priority-EDF), the GA, and an ALNS-style destroy-and-repair method are the non-learning baselines. GA and ALNS-style candidate settings were examined on validation requests using schedule feasibility, final planning quality, and computation time. We fixed one problem-specific configuration for each method before the reported test evaluation, selecting a practical quality-time balance rather than optimizing either measure alone. During development, GA order crossover was streamlined to avoid repeated set construction, while ALNS destruction avoided repeated position-set construction and recalculation of an unchanged priority median. The submitted GA used population 100, at most 500 generations, order crossover probability 0.9, swap mutation probability 0.05, five elites, tournament size three, and patience 50. The submitted ALNS-style solver used random-position, low-priority and near-median-priority destruction in a fixed cycle, priority/deadline-oriented repair, Metropolis acceptance, destruction fraction 0.20, initial temperature 5.0, cooling factor 0.997, at most 2000 iterations, and patience 300; CP-SAT repair was disabled. We use ALNS-style because operator-selection weights were not adapted. The selected settings offer a useful balance for the tested problem; a finite validation search cannot establish a global parameter optimum. Appendix A Table A2 lists the fixed settings.

2.3.4. Ablations and Implementation Diagnostic

Ablations retain the dataset split, training episodes, candidate set, and hard constraints. MLP-PPO removes graph message passing; the conflict-only variant retains R1 alone; and the fixed-weight variant disables adaptive Lagrange updates. Dynamic full-graph reconstruction does not change the decision policy and is used only to diagnose graph maintenance time, not as primary evidence for the relation or constraint mechanisms. The fixed-weight control starts from the same initial emergency and retention multipliers (1.0 each) as the adaptive model and holds them constant. This comparison isolates the update rule at that reference pair; it does not test every possible fixed-weight setting.

2.3.5. Metrics and Timing Definitions

Metrics are grouped into quality, service, stability, and timeliness. The normalized score J in Equation (2) and regular-grid coverage describe schedule quality. Emergency service is measured by emergency-unit satisfaction and compliance with $C^E \geq 0.85q^E$. Plan stability is measured by weighted original-plan retention and the perturbation measures in Equations (10) and (11). Timeliness is measured by dynamic response time and the 300 s timing contract.

Preprocessing includes request and candidate loading, strip construction, and graph building. Static planning prepares the regular-task plan before emergency arrival. The recorded method-specific post-arrival response interval begins after arrival candidates have been materialized and the method-specific replanning call starts; it does not include the full task from initial preprocessing. A complete per-request online cost includes preprocessing, static pre-planning, emergency materialization, dynamic replanning, and final feasibility processing for every method. Offline GCN-PPO training and offline GA/ALNS parameter selection are outside that per-request cost. The reported two-stage within_budget field tests separate static and post-arrival limits and is not a single end-to-end deadline or a feasibility flag. Cross-method timing statements below identify the particular recorded interval or two-stage criterion used.

For a nominal post-arrival budget B, the two-stage timing check uses T_static $\leq$ 20 + delta(B) and T_dynamic $\leq$ B + delta(B), where delta(B) = max(0.02 s, 0.02B). At B = 300 s, the thresholds are 26 s and 306 s; the stage times are checked separately rather than added into a 300 s total. We therefore show static and post-arrival compliance separately when interpreting the reported joint percentages. Equivalently, the stored indicator is within_budget(B) = 1 if both T_static $\leq$ 20 s + delta(B) and T_dynamic $\leq$ B + delta(B) and 0 otherwise.

2.3.6. Training, Statistics, and Reproducibility

The default encoder has three relation-specific layers, each with five relation channels. Its dimensions are 13 $\rightarrow$ 128 $\rightarrow$ 128 $\rightarrow$ 32, with zero dropout, residual projection, layer normalization, maximum aggregation within each relation, and summation across relations. PPO uses a discount factor of 0.99, a GAE parameter of 0.95, and a clipping coefficient of 0.2. Learning rates for the policy, value heads, and encoder are 1×10^{-4}, 3×10^{-4}, and 1×10^{-4}, respectively. Each update uses six epochs, a batch size of 64, a buffer of 2048 transitions, a 512-step trajectory budget, and a KL threshold of 0.02.

Emergency-unit, emergency-logical, and weighted-retention targets are $0.85q^E$, $0.80q^L$, and 0.45, with a maximum perturbation ratio of 0.60. Training lasts 784 episodes, validation is performed every 56 episodes, and checkpoints are saved every 28 episodes. The action hierarchy retains at most 128 demand groups, four opportunities per group, and 512 candidates globally. Reward weights for regular coverage, emergency units, emergency logical tasks, earliness, and retention are 1, 6, 3, 0.25, and 0.35. All settings were fixed before evaluation on the held-out test set.

Training uses seeds 11, 23, 42, 67, and 89. Every seed starts from an independent random initialization, trains for 784 episodes on the same 300 requests, and selects checkpoints every 56 episodes on the same 50 validation requests. Reward and constraint weights were fixed before evaluation on the held-out test set, based on emergency priority and single-seed validation records.

Descriptive means are computed over 750 evaluations. For 95% confidence intervals and paired method comparisons, the five seeds are first averaged within each request, after which the 150 requests are treated as independent units. Load-shock tests use 75 paired scenario families. The quality-time sensitivity analysis also reports the proportion of same-

request, same-seed pairs in which GCN-PPO reaches 85% or 90% of the natural-termination ALNS return, together with conditional speedup among qualifying pairs.

Continuous metrics are summarized by means, quantiles, and 95% confidence intervals, and method differences are paired by request. Scale-load strata are formed after averaging seeds within a request. Spearman analysis uses 150 requests, with candidate-count quartiles of approximately 1056, 1539, 2222, and 2986. Independent five-seed training, per-request records, and code and configuration hashes close the experimental contract [42,43].

The methods were evaluated in the same CPU-only environment running Ubuntu 22.04 (glibc 2.35), with 256 GiB RAM, Python 3.10.12, PyTorch 2.13.0 + cpu, and PyG ≥ 2.3.0 specified as a dependency lower bound; each process used one CPU thread. The experiments were run on a cloud server; the CPU model was not specified in the available records. Offline policy training was excluded from per-request online time; each seed used 784 episodes, approximately 19.5-21.1 h per seed (seed 11: 75,860 s).

$$\bar{x} = \frac{1}{n}\sum x_i, s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n-1}}, \mathrm{CI}_{95\%} = \bar{x} \pm t_{0.975,n-1} \cdot \frac{s}{\sqrt{n}} \quad (12)$$

OpenAI Codex (OpenAI, San Francisco, CA, USA) was used as an assistive tool for code review and modification and for preparing workflow diagrams and result visualizations. All source-code changes were reviewed by the authors and validated through unit tests, engineering audits, and reproduction of the reported outputs. The authors retained responsibility for the experimental design, execution, numerical analysis, interpretation, and final figures.

Figure 4 summarizes the experimental control sequence. Engineering checks and a validation-only single-seed diagnostic precede the formal five-seed runs. The held-out test split is evaluated only after settings and checkpoints have been fixed, and accepted records are aggregated by request for the reported statistical comparisons.

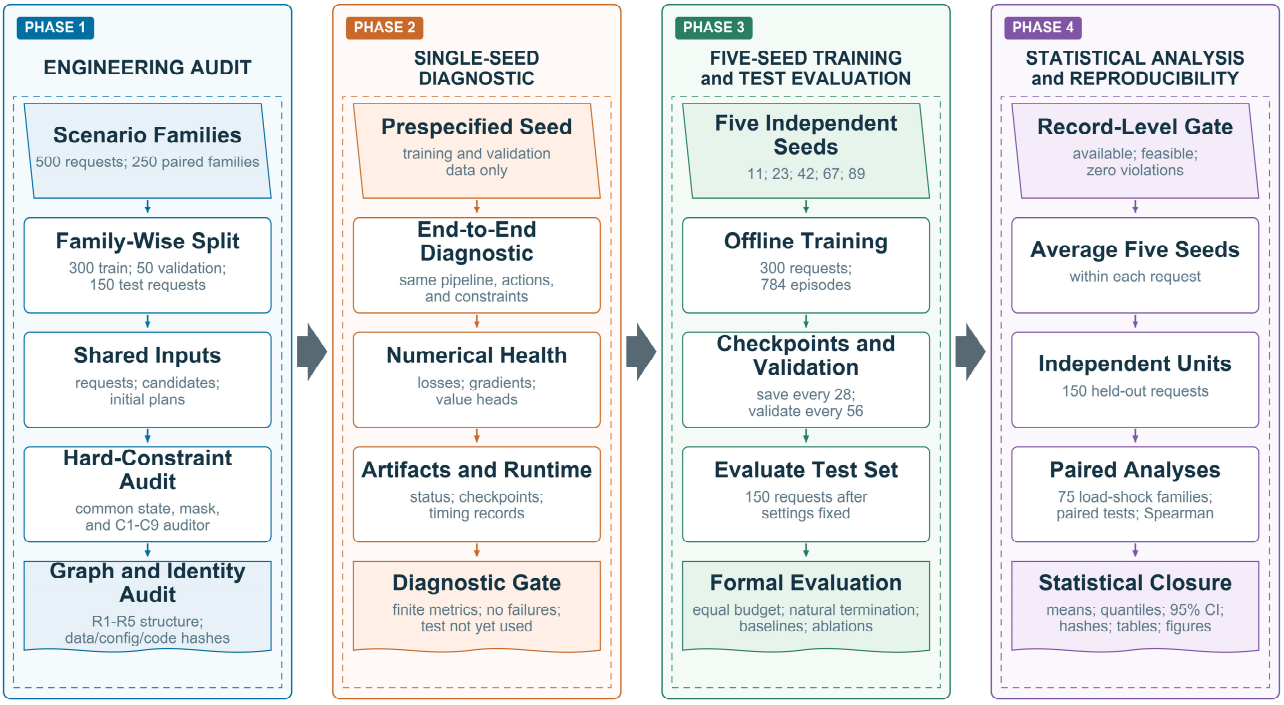


Figure 4. Audited training and evaluation workflow. Engineering checks and a validation-only single-seed diagnostic precede independent five-seed training; the held-out test set is evaluated after settings are fixed, and inference records pass feasibility gates before request-level statistical aggregation.

3. Results

3.1. Engineering Consistency, Data Integrity, and Feasibility

All records used in the main evaluation passed the availability, feasibility, and hard-constraint gates. Across the 750 GCN-PPO test evaluations, the number of C1-C9 violations was zero, and the relation-structure audit passed in all 750 cases. The request, model, configuration, and raw-record hashes were stored in the evaluation manifest, so subsequent differences cannot be attributed to changes in the candidate set or constraint auditor.

3.2. Performance Under a 300 s Post-Arrival Allowance

In this 300 s group, the recorded two-stage flag requires static planning within 26 s and post-arrival replanning within 306 s. Table 2 reports that joint indicator for the fixed settings. Section 2.3.3 and Appendix A Table A2 specify the GA and ALNS-style settings used.

Table 2. Results for the fixed configurations (750 runs per method). The timing column reports the joint static and post-arrival stage indicator.

Method	Two-Stage Timing Pass (%)	Perturbation	Regular (%)	Retention (%)	Emergency (%)	Planning Quality J [95% CI]
Priority-EDF	100.0	5.713	68.01	36.37	70.42	72.67 [70.76, 74.59]
GCN-PPO	96.4	4.186	73.97	39.27	69.25	72.29 [69.92, 74.66]
ALNS	53.1	4.843	77.14	40.35	79.60	82.16 [80.38, 83.94]
GA	0.0	4.541	77.56	42.07	79.39	82.07 [80.29, 83.86]

All 750 GCN-PPO schedules were feasible and free of hard-constraint violations, and 723/750 met the recorded two-stage flag. ALNS-style met the joint two-stage timing criterion in 398/750 evaluations (53.1%), whereas the GA met it in 0/750 evaluations (0%). The ALNS-style and GA configurations had mean scalar J values of 82.16 and 82.07, above the GCN-PPO mean of 72.29. The methods thus occupy different positions in the observed quality-time trade-off. Relative to Priority-EDF, GCN-PPO had J = 72.29 versus 72.67, 5.96 percentage points more regular coverage, 2.90 points more weighted retention, and 26.7% less perturbation cost. The 45% retention reference is a training target rather than a hard per-instance constraint.

The scale-dependent limitation is explicit in the natural-termination diagnostics. For the 36-satellite/medium-load cell, 22 requests across five seeds give 110 GCN-PPO records; 69/110 (62.7%) had post-arrival response within 300 s. The recorded means were preprocessing 10.58 s, static planning 12.98 s, dynamic response 286.42 s and online end-to-end 312.81 s. Graph update 60.23 s and policy scoring 230.35 s are nested diagnostic accumulators, not mutually exclusive durations to add to the outer timer. The negative GCN-PPO quality difference under 9-satellite high load is likewise retained as an operating-regime limitation.

3.3. Natural-Termination Timeliness, Emergency Constraint, and Low Perturbation

Table 3 summarizes the natural-termination response time and perturbation, and Figure 5 shows the response-time distribution. Here, natural-termination response is the recorded post-arrival interval; it is distinct from the two-stage timing flag in Table 2 and does not include all regular-task pre-planning. GCN-PPO's mean recorded post-arrival response was 102.4 s, 9.35% of the 1095.4 s mean for the ALNS-style implementation; the corresponding medians were 64.7 s and 608.0 s. GCN-PPO's post-arrival interval was within 300 s in 93.6% of evaluations. Every 9- and 18-satellite case met this limit. For the 36-satellite low-load, medium-load, and overall subsets, the rates were 93.0%, 62.7%, and 77.1%, respectively. The target $C^E \geq 0.85q^E$ was met in 736/750 evaluations (98.1%), and

the mean C^E/q^E ratio was 1.076. Relative to MLP-PPO and ALNS, per-edit perturbation was 31.4% and 35.9% lower, while size-normalized perturbation was 34.8% and 34.7% lower.

Table 3. Natural-termination post-arrival response and perturbation (fixed method configurations; 750 evaluations).

Method	Mean Response (s)	Median (s)	p_{95} (s)	≤300 s (%)	Per-Edit Perturbation	Size-Normalized Perturbation
Priority-EDF	7.9	6.3	21.4	100.0	0.184	0.114
MLP-PPO	92.7	61.6	278.7	95.5	0.144	0.088
ALNS	1095.4	608.0	3896.3	25.5	0.154	0.087
GCN-PPO	102.4	64.7	317.1	93.6	0.099	0.057

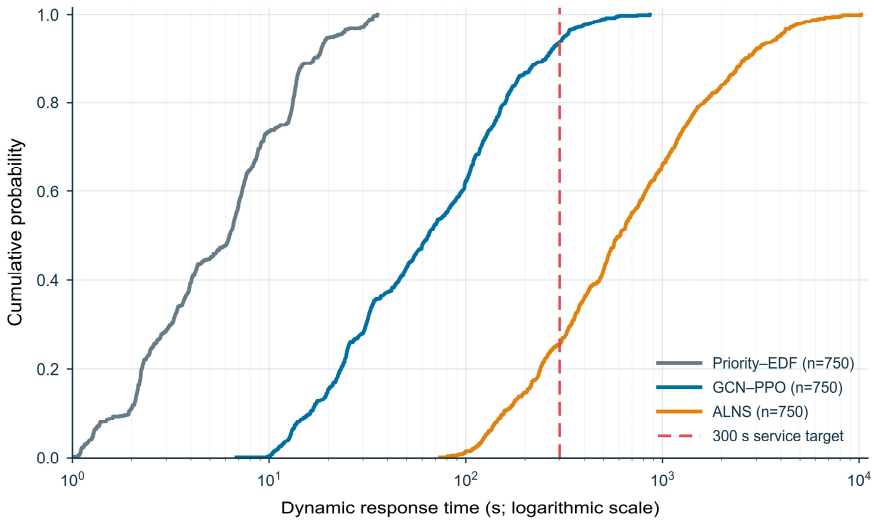


Figure 5. Cumulative distribution of dynamic response time (logarithmic horizontal axis; 150 requests × 5 seeds). The red line marks the prespecified 300 s service target.

3.4. Stability Contributions of Graph Relations and Adaptive Constraints

Table 4 reports the relation and constraint ablations. The full method produced smaller and more repeatable plan edits. Compared with MLP-PPO, it achieved lower perturbation cost and higher weighted original-plan retention in all 16 cells defined by satellite scale, task count, and materialized load. The five-relation model reduced perturbation relative to the conflict-only model in all 16 cells, and adaptive multipliers improved both retention and perturbation over fixed weights in all 16 cells. The request-level cross-seed standard deviation of test perturbation decreased from 0.675 for MLP-PPO to 0.464 for the full method. The advantage therefore concerns repeatability as well as the mean perturbation. Figure 6 visualizes the corresponding perturbation intensity and cross-seed variability.

Table 4. Natural-termination relation and constraint ablations (means: 750 evaluations; stratum wins: 16 satellite-scale/task-count/materialized-load cells).

Variant	Planning Quality	Retention (%)	Perturbation	Cross-Seed SD of Test Perturbation	Full Method Relative to Variant
Full GCN-PPO	73.91	39.03	4.499	0.464	Reference
MLP-PPO	77.41	35.91	5.496	0.675	Lower perturbation in 16/16; higher retention in 16/16
Conflict-only	76.03	38.25	4.837	0.756	Lower perturbation in 16/16
Fixed-weight	74.00	36.99	4.918	0.682	Both perturbation and retention improved in 16/16

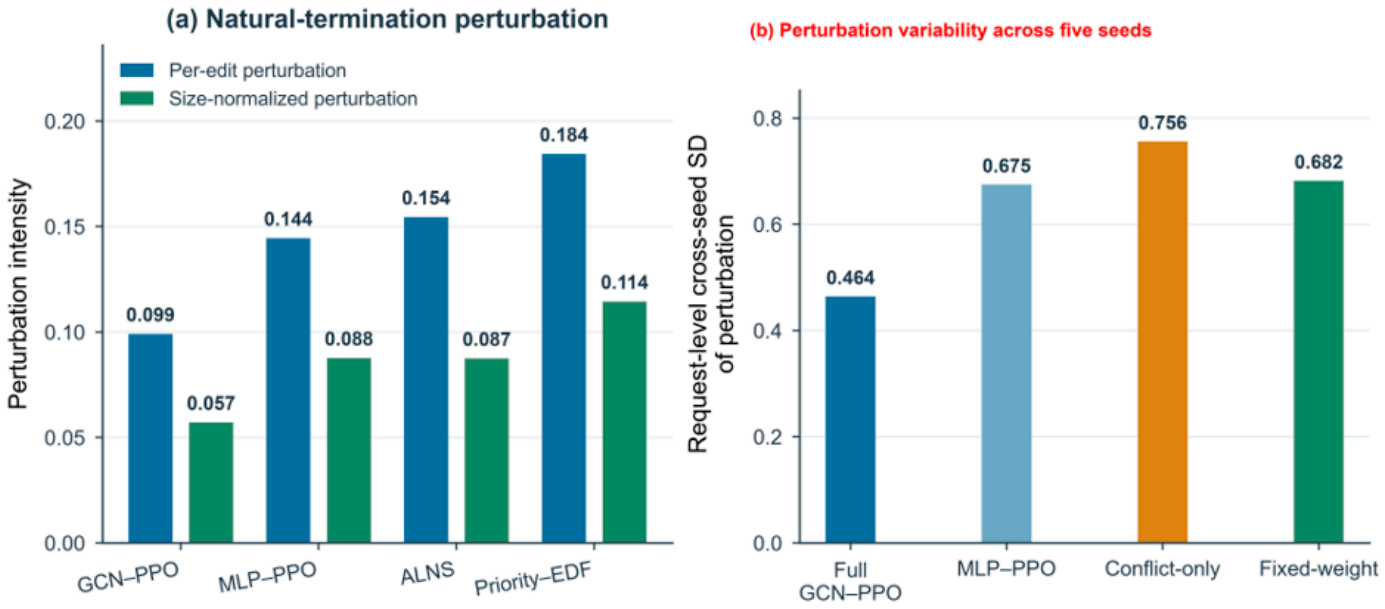


Figure 6. Perturbation intensity and perturbation variability across five seeds for 150 held-out test requests.

The 16 strata are defined by satellite scale, logical-task count, and materialized load band. The request/evaluation counts are 9/M/202 (11/55), 9/M/204 (11/55), 9/H/402 (13/65), 9/H/404 (13/65); 18/L/204 (5/25), 18/L/208 (5/25), 18/M/404 (11/55), 18/M/408 (11/55), 18/H/804 (14/70), 18/H/808 (14/70); 36/L/208 (5/25), 36/L/216 (5/25), 36/L/408 (5/25), 36/L/416 (5/25), 36/M/808 (11/55), and 36/M/816 (11/55). The second number is five seeds times the request count, and all comparisons use paired request-seed cells. The paired ablation differences for these strata are reported in Appendix A Table A3.

3.5. Load Shock, Scale-Dependent Advantage, and the Value of Time

In the 16 detailed cells, GCN-PPO had quality advantages of 2.25 to 5.12 in the 18-satellite/804-task, 18-satellite/808-task, 36-satellite/208-task, and 36-satellite/408-task cells ($p < 0.05$). In the seven coarse cells of Table 5, the quality difference was positive in six cells and negative only under 9-satellite high load. High-load regular coverage improved by 6.7 to 19.3 percentage points. From proportional to high load, perturbation increased by 2.791 for GCN-PPO, compared with 3.292 for MLP-PPO, 3.827 for ALNS, and 3.535 for Priority-EDF. GCN-PPO had the smaller increase in 64, 73, and 63 of the 75 paired families against these three methods, respectively (paired $p < 3 \times 10^{-10}$). As candidate-count quartiles increased from about 1056 to 2986, the MLP-PPO minus GCN-PPO perturbation gap widened from 0.60 to 1.48 (Spearman $\rho = 0.616$, $n = 150$, $p < 5 \times 10^{-17}$). In the reported natural-termination configuration, ALNS-style gained 9.66 J points relative to GCN-PPO and used 993.0 s more recorded post-arrival time on average.

Against the fixed natural-termination ALNS-style return on matched request-seed pairs, GCN-PPO attained at least 85% and 90% of that return in 536/750 (71.5%) and 275/750 (36.7%) pairs. Conditional time ratios among only the qualifying pairs ranged from about 7 to 17 and 6 to 18 across scale-load cells. These subset-conditioned ratios describe the observed trade-off for the reported settings. Figure 7 summarizes the 300 s timing-contract pass rates and weighted perturbation under the fixed-configuration protocol.

Table 5. GCN-PPO relative to Priority-EDF across seven scale-load groups (n in cells denotes independent requests; the post-arrival ≤ 300 s percentages use five seeds per request, e.g., 69/110 for 36-satellite medium load). Quality difference equals GCN-PPO planning quality minus Priority-EDF planning quality; perturbation advantage equals Priority-EDF perturbation minus GCN-PPO perturbation. Positive values favor GCN-PPO.

Scale	Load	Quality Difference	Perturbation Advantage	Advantage	GCN-PPO ≤ 300 s (%)
9	Medium	+3.23	+0.56	Quality and perturbation	100.0 (n = 22)
9	High	−3.29	+0.72	Lower perturbation	100.0 (n = 26)
18	Low	+1.49	+1.11	Quality, retention, and perturbation	100.0 (n = 10)
18	Medium	+1.55	+1.20	Quality, retention, and perturbation	100.0 (n = 22)
18	High	+2.72	+1.14	High-load coverage	100.0 (n = 28)
36	Low	+2.83	+1.86	Quality, emergency service, and perturbation	93.0 (n = 20)
36	Medium	+0.82	+2.09	Largest perturbation advantage	62.7 (n = 22)

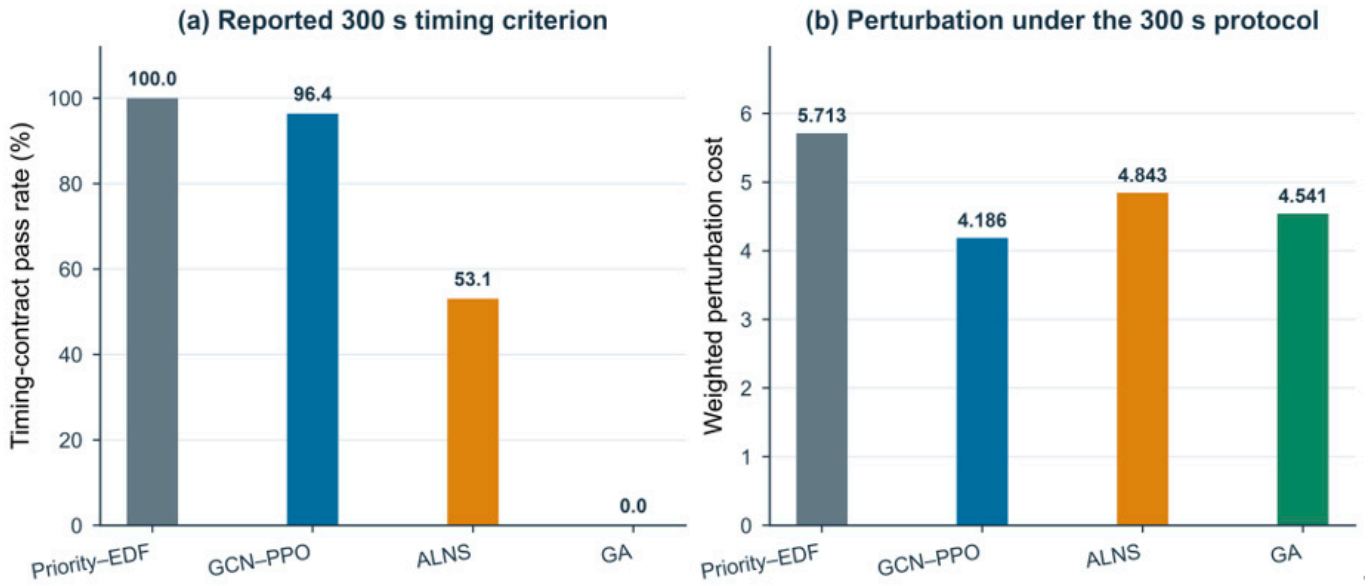


Figure 7. Preregistered timing-contract pass rate and weighted perturbation cost under the 300 s equal-budget protocol (150 requests × 5 seeds). The timing contract includes a 2% logging tolerance; every reported schedule passed feasibility and hard-constraint audits.

4. Discussion

Priority-EDF completes one rule-based construction and is therefore the fastest baseline, but its local priority and deadline decisions cannot jointly evaluate cross-satellite substitution, strip merging, and future opportunity scarcity. ALNS and the GA can improve solution quality through repeated neighborhood repair or population evolution [13–18], yet every new emergency request triggers many rounds of candidate decoding and constraint checking. This difference explains why their quality gains come with substantially longer online response times.

MLP-PPO scores individual candidates and the global state without explicitly representing whether two opportunities conflict, substitute for one another, or form a continuous strip. Separating these meanings into five message channels helps the policy locate the part of the plan that requires editing. The protection set then locks original strips that do not obstruct the last feasible emergency opportunity, and the policy acts only on conflicting strips and still-open regular or emergency opportunities. This combination accounts for the consistent reductions in perturbation and cross-seed variability.

The advantage is workload-dependent. In the 36-satellite/medium-load group, 69/110 GCN-PPO runs (62.7%) returned within 300 s under natural termination, compared with 93.6% overall. Mean post-arrival response was 286.42 s; the recorded graph-update and policy-scoring diagnostics were 60.23 and 230.35 s, respectively. These nested diagnostics cannot be added to obtain a response-time breakdown but indicate substantial policy-scoring cost at this scale. At nine satellites under high load, mean planning quality J was 3.29 points below Priority-EDF. These groups delimit the average advantages reported above.

GCN-PPO learns relation patterns and action order offline. During online planning, it constructs the schedule by selecting only actions admitted by the hard-constraint mask and uses feasibility-normalized constraints to match service pressure to the resources of the current request. This avoids a long request-specific search while preserving executability. For floods, earthquakes, and similar emergency observations, a feasible schedule delivered before plan upload and the next useful pass can have greater operational value than a higher-quality schedule that arrives after that decision point.

The conclusions apply to centralized 24 h planning in which a regular schedule already exists and a batch of emergency requests arrives before schedule upload. The test set contains task identifiers absent from training, but it is not a geographic block holdout; the generalization claim is therefore limited to new task combinations and resource scales.

As a concrete example from the reported test evaluations, SCN-00211 is an 18-satellite medium-load request with 404 logical tasks. In the seed-11 record, the original plan had 68 pending strips, of which 28 were retained after emergency insertion. The resulting schedule served 281/400 emergency completion units and 70/100 regular logical tasks, with planning quality J = 72.498447, perturbation cost 3.885263, recorded post-arrival response of 58.941736 s, and zero C1-C9 hard-constraint violations. This request is one of the evaluations summarized in Tables 2–5 and illustrates the observed balance among task service, schedule stability, and response time. The five relation types in Figure 3 define its opportunity-graph representation, while the action mask and protection rule constrain feasible local edits.

5. Conclusions

This study developed a multi-relational GCN-PPO method for 24 h multi-satellite SAR emergency replanning. The five-relation graph distinguishes conflict, temporal sequence, cross-satellite competition, demand alternatives, and strip synergy. A three-level variable-length policy handles changing candidate sets, while feasibility-normalized constraints control emergency-service shortfall and original-plan perturbation. After an emergency arrives, the protection set retains unrelated original strips and limits replanning to the local conflicts that affect emergency feasibility. Across five seeds and 150 held-out requests, GCN-PPO returned a feasible schedule in every evaluation. It passed the 300 s timing contract in 96.4% of evaluations, compared with 53.1% for ALNS and 0% for GA. Under natural termination, its mean response time was 102.4 s, equal to 9.35% of the ALNS time. The normalized emergency target was met in 98.1% of evaluations, and per-edit perturbation was 31.4% lower than that of MLP-PPO and 35.9% lower than that of ALNS. The conclusions remain limited to the simulated 9-36-satellite workloads, trained constellation sizes, and pre-upload batch-arrival setting evaluated here.

Future work will examine larger SAR constellations, broader geographic hold-outs, and a wider validation search over GA/ALNS operators and parameters using common complete-task timing. Distributed state observation and uncertainty-aware allocation [44,45] suggest further directions, but the current planner is centralized and does not implement those mechanisms.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/rs18213675/s1, Supplementary File S1 (Frozen Task Scenarios) contains fixed task requests, scenario specifications and manifests, the source workbook, the shared SGP4 visibility SQLite database, selected scenario-generation source snapshots, boundary provenance, a SHA-256 manifest, and an independent integrity validator.

Author Contributions: Conceptualization, X.W. and C.S.; methodology, X.W.; software, X.W.; validation, X.W.; formal analysis, X.W.; data curation, X.W.; writing—original draft preparation, X.W.; writing—review and editing, C.S., L.B., and X.W.; visualization, X.W.; supervision, C.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China, grant number 2023YFB3906103. This research was also supported by the 2025 Postgraduate Research and Practice Innovation Program of Jiangsu Province (Grant No. KYCX25_1626), under the project "Research on Key Technologies for Improving the Timeliness of Remote Sensing Satellite Data Reception.".

Institutional Review Board Statement: Not applicable. This simulation-based algorithm study involved neither human participants nor animals.

Informed Consent Statement: Not applicable.

Data Availability Statement: Supplementary File S1 is publicly available with this article. It provides 500 fixed task requests (300 training, 50 validation, and 150 held-out test), the source workbook, fixed SGP4 visibility SQLite database, selected scenario-generation scripts, a SHA-256 manifest, and an independent validator for file integrity and key scenario metadata. The package documents and permits inspection of the scenario inputs. The generator scripts describe the sampling and materialization workflow and require the study's scheduling dependencies for exact reconstruction of the same requests. Provenance for the DataV boundary source is documented in the package.

Acknowledgments: During the preparation of this manuscript, the authors used OpenAI Codex for translation, language editing, Word-template formatting, code review and modification, and preparation of workflow diagrams and result visualizations. The authors reviewed and validated all source-code changes, figures, numerical results, and resulting text and take full responsibility for the content of the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

ALNS	Adaptive large neighborhood search
CI	Confidence interval
DQN	Deep Q-network
EDF	Earliest deadline first
GA	Genetic algorithm
GAE	Generalized advantage estimation
GCN	Graph convolutional network
GCN-PPO	Multi-relational graph convolutional network with constrained proximal policy optimization
GPU	Graphics processing unit
KL	Kullback-Leibler
MLP	Multilayer perceptron
MLP-PPO	Multilayer perceptron with proximal policy optimization
PPO	Proximal policy optimization
RGCN	Relational graph convolutional network
SAR	Synthetic aperture radar

SD Standard deviation
SGP4 Simplified General Perturbations 4

Appendix A

Note A1. The paired 24 h test evaluation used 60 shards, seeds 11/23/42/67/89, and methods Priority-EDF, GA, ALNS-style, and GCN-PPO. Post-arrival budgets are 5/20/60/300 s, and the nominal static service target is 20 s. Requests, candidates, and constraint audits are paired by request and seed. Scenario dataset SHA-256: 98165c5b9bf09721c9d9db1fddadfd792ebc4f8cd23c7b30d419784609be242c.

Note A2. The ninth node feature is a remaining-day proxy: for emergency strips, m_day = clip [1 − (start_epoch mod 86400)/86400, 0, 1], whereas regular strips take 1. For feature 12, c_i counts same-satellite candidate strips whose imaging intervals overlap strip i. For feature 13, x_i counts different-satellite candidates that share at least one logical target key with strip i. Each count is normalized within the request as log1p(c_i)/log1p(max(c_max, 1)) or log1p(x_i)/log1p(max(x_max, 1)), respectively, and then clipped to [0, 1]. These structural counts are distinct from the R1 and R3 edge-construction tests. The four pair diagnostics are two continuous measures (interval-overlap ratio and a planar longitude/latitude radian-distance proxy) and two binary flags (look-side switch and same satellite); discrete R1-R5 rules construct edges, and relation type selects the RGCN message channel. Let k_i be the number of distinct target keys covered by strip i and k_max the request maximum; the coverage feature is y_cover = log1p(k_i)/max(log1p(k_max), 1×10^{-9}). For intervals [s_i, e_i) and [s_j, e_j), r_time = max(0, min(e_i, e_j) − max(s_i, s_j))/max(min(max(e_i − s_i, 1 s), max(e_j − s_j, 1 s)), 1 s). With coordinate differences in radians, r_space = 1 − clip(sqrt((delta_lon)^2 + (delta_lat)^2)/pi, 0, 1). The binary flag b_side is 1 only for nonzero opposite-look sides, and b_sat is 1 for equal satellite IDs; both are 0 otherwise. All four diagnostics lie in [0, 1].

Note A3. Statistical and stratum protocol. Descriptive means and 95% confidence intervals use request-level values after averaging five seeds within each request. Paired comparisons use the same request, and load-shock comparisons use 75 paired scenario families. The 16 paired strata and their request/evaluation counts are listed after Table 4.

Table A1. Deterministic relation construction rules; orbit capacities remain decoder constraints rather than R1 edges.

Relation	Direction	Construction Rule
R1	both	Same satellite; overlapping half-open intervals or infeasible opposite-look gap <130 s.
R2	earlier to later	Same satellite and orbit; feasible transition; earlier-start to later-end span ≤ 240 s.
R3	both	Different satellites; great-circle separation < 3 degrees and start-time difference < 30 min.
R4	both	At least one shared logical completion unit in Q(v).
R5	both	Same satellite/orbit/look, disjoint logical keys, valid merged metadata, span ≤ 240 s, and a candidate containing both metadata sets.

These settings define the GA and ALNS-style implementations in Tables 2 and 3. They were selected as a practical balance among the validation settings considered; the search does not establish global parameter optimality.

Table A2. GA and ALNS-style settings fixed for the submitted evaluation after validation-set selection based on feasible planning quality and computation time.

Method	Parameter Group	Submitted Setting
GA	Search	Population 100; at most 500 generations; patience 50
GA	Operators	Order crossover 0.9; swap mutation 0.05; five elites; tournament size three
ALNS-style	Search	At most 2000 iterations; destruction fraction 0.20; patience 300
ALNS-style	Operators	Three destruction operators in fixed cycle; priority/deadline repair; Metropolis T0 = 5, cooling 0.997; CP-SAT off

Table A3. Detailed held-out test strata. Evaluation count equals five seeds per independent request. The final two columns report paired stratum mean differences from the same test requests and five seeds: delta perturbation is each ablation minus full GCN-PPO (MLP-PPO/conflict-only/fixed-weight); delta weighted retention is full GCN-PPO minus each ablation (MLP-PPO/fixed-weight), in percentage points. Positive values mean the full model has lower perturbation or higher weighted retention.

Scale	Load	Logical Tasks	Requests	Evaluations	Delta Perturbation M/R/F	Delta Weighted Retention (pp) M/F
9	M	202	11	55	0.393/0.124/0.175	4.679/3.506
9	M	204	11	55	0.879/0.434/0.356	4.740/3.960
9	H	402	13	65	0.454/0.186/0.166	2.824/1.420
9	H	404	13	65	0.802/0.407/0.325	3.171/2.095
18	L	204	5	25	0.772/0.167/0.333	3.343/3.023
18	L	208	5	25	0.964/0.368/0.482	1.255/1.441
18	M	404	11	55	0.744/0.336/0.459	3.220/3.723
18	M	408	11	55	1.199/0.439/0.241	4.221/0.641
18	H	804	14	70	0.914/0.294/0.354	4.645/2.551
18	H	808	14	70	1.353/0.350/0.407	3.509/1.480
36	L	208	5	25	0.950/0.157/0.577	1.905/1.543
36	L	216	5	25	1.198/0.369/0.756	0.464/1.217
36	L	408	5	25	0.947/0.234/0.517	1.113/1.424
36	L	416	5	25	1.573/0.607/0.580	1.371/0.706
36	M	808	11	55	1.037/0.208/0.494	1.416/0.640
36	M	816	11	55	2.060/0.684/0.959	2.496/1.858

References

1. Zhao, Q.; Yu, L.; Du, Z.; Peng, D.; Hao, P.; Zhang, Y.; Gong, P. An Overview of the Applications of Earth Observation Satellite Data: Impacts and Future Trends. *Remote Sens*. **2022**, *14*, 1863. [CrossRef]
2. Ke, Z.F.; Huang, S.S.; Li, Y.L.; Qiao, K.; Teng, F.; Ruan, H.; Wang, X.T.; Wei, C.Q.; Ma, X.L. Research on the Development Status and Key Technologies of Large LEO Remote Sensing Constellations. *Spacecr. Recovery Remote Sens*. **2023**, *44*, 93–101. (In Chinese) [CrossRef]
3. Wang, X.; Wu, G.; Xing, L.; Pedrycz, W. Agile Earth Observation Satellite Scheduling over 20 Years: Formulations, Methods, and Future Directions. *IEEE Syst. J*. **2021**, *15*, 3881–3892. [CrossRef]
4. Du, Y.H.; Xing, L.N.; Yao, F.; Chen, Y.G. Survey on Models, Algorithms and General Techniques for Spacecraft Mission Scheduling. *Acta Autom. Sin*. **2021**, *47*, 2715–2741. (In Chinese) [CrossRef]
5. Xiang, S.; Chen, Y.G.; Li, G.L.; Xing, L.N. Review on Satellite Autonomous and Collaborative Task Scheduling Planning. *Acta Autom. Sin*. **2019**, *45*, 252–264. (In Chinese) [CrossRef]

6. Wolfe, W.J.; Sorensen, S.E. Three Scheduling Algorithms Applied to the Earth Observing Systems Domain. *Manag. Sci.* **2000**, *46*, 148–166. [CrossRef]
7. Lemaître, M.; Verfaillie, G.; Jouhaud, F.; Lachiver, J.M.; Bataille, N. Selecting and Scheduling Observations of Agile Satellites. *Aerosp. Sci. Technol.* **2002**, *6*, 367–381. [CrossRef]
8. Bianchessi, N.; Righini, G. Planning and Scheduling Algorithms for the COSMO-SkyMed Constellation. *Aerosp. Sci. Technol.* **2008**, *12*, 535–544. [CrossRef]
9. Peng, G.; Dewil, R.; Verbeeck, C.; Gunawan, A.; Xing, L.; Vansteenwegen, P. Agile Earth Observation Satellite Scheduling: An Orienteering Problem with Time-Dependent Profits and Travel Times. *Comput. Oper. Res.* **2019**, *111*, 84–98. [CrossRef]
10. Peng, G.; Song, G.; Xing, L.; Gunawan, A.; Vansteenwegen, P. An Exact Algorithm for Agile Earth Observation Satellite Scheduling with Time-Dependent Profits. *Comput. Oper. Res.* **2020**, *120*, 104946. [CrossRef]
11. Wu, G.; Luo, Q.; Du, X.; Chen, Y.; Suganthan, P.N.; Wang, X. Ensemble of Metaheuristic and Exact Algorithm Based on the Divide-and-Conquer Framework for Multisatellite Observation Scheduling. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 4396–4408. [CrossRef]
12. Wang, J.; Song, G.; Liang, Z.; Demeulemeester, E.; Hu, X.; Liu, J. Unrelated Parallel Machine Scheduling with Multiple Time Windows: An Application to Earth Observation Satellite Scheduling. *Comput. Oper. Res.* **2023**, *149*, 106010. [CrossRef]
13. Liu, X.; Laporte, G.; Chen, Y.; He, R. An Adaptive Large Neighborhood Search Metaheuristic for Agile Satellite Scheduling with Time-Dependent Transition Time. *Comput. Oper. Res.* **2017**, *86*, 41–53. [CrossRef]
14. He, L.; Liu, X.; Laporte, G.; Chen, Y.; Chen, Y. An Improved Adaptive Large Neighborhood Search Algorithm for Multiple Agile Satellites Scheduling. *Comput. Oper. Res.* **2018**, *100*, 12–25. [CrossRef]
15. Ropke, S.; Pisinger, D. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transp. Sci.* **2006**, *40*, 455–472. [CrossRef]
16. Niu, X.; Tang, H.; Wu, L. Satellite Scheduling of Large Areal Tasks for Rapid Response to Natural Disaster Using a Multi-Objective Genetic Algorithm. *Int. J. Disaster Risk Reduct.* **2018**, *28*, 813–825. [CrossRef]
17. Wu, G.H.; Du, X.; Wang, X.W.; Chen, X.J.; Zhang, W.F.; Mao, C.H. Multi-Satellite Emergency Scheduling Algorithm Considering Dynamic Selection of Neighborhood Structure. *Control Decis.* **2022**, *37*, 1685–1694. (In Chinese) [CrossRef]
18. Li, X.T.; Wang, L.; Chen, Y.W.; Xing, L.N.; Chen, Y.G. Multi-Scenario Multi-Satellite Mission Planning Method Based on Adaptive Large Neighborhood Search. *J. Syst. Simul.* **2025**, *37*, 1836–1847. (In Chinese) [CrossRef]
19. Huang, Y.; Mu, Z.; Wu, S.; Cui, B.; Duan, Y. Revising the Observation Satellite Scheduling Problem Based on Deep Reinforcement Learning. *Remote Sens.* **2021**, *13*, 2377. [CrossRef]
20. Wei, L.N.; Chen, Y.N.; Chen, M.; Chen, Y.W. Deep Reinforcement Learning and Parameter Transfer Based Approach for the Multi-Objective Agile Earth Observation Satellite Scheduling Problem. *Appl. Soft Comput.* **2021**, *110*, 107607. [CrossRef]
21. Li, D.; Wang, H.; Yang, Z.; Gu, Y.; Shen, S. An Online Distributed Satellite Cooperative Observation Scheduling Algorithm Based on Multiagent Deep Reinforcement Learning. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 1901–1905. [CrossRef]
22. He, Y.; Xing, L.; Chen, Y.; Pedrycz, W.; Wang, L.; Wu, G. A Generic Markov Decision Process Model and Reinforcement Learning Method for Scheduling Agile Earth Observation Satellites. *IEEE Trans. Syst. Man. Cybern. Syst.* **2022**, *52*, 1463–1474. [CrossRef]
23. Shi, Q.; Li, L.; Fang, Z.; Bi, X.; Liu, H.; Zhang, X.; Chen, W.; Yu, J. Efficient and Fair PPO-Based Integrated Scheduling Method for Multiple Tasks of SATech-01 Satellite. *Chin. J. Aeronaut.* **2024**, *37*, 417–430. [CrossRef]
24. Song, Y.; Ou, J.; Pedrycz, W.; Suganthan, P.N.; Wang, X.; Xing, L.; Zhang, Y. Generalized Model and Deep Reinforcement Learning-Based Evolutionary Method for Multitype Satellite Observation Scheduling. *IEEE Trans. Syst. Man. Cybern. Syst.* **2024**, *54*, 2576–2589. [CrossRef]
25. Chun, J.; Yang, W.; Liu, X.; Wu, G.; He, L.; Xing, L. Deep Reinforcement Learning for the Agile Earth Observation Satellite Scheduling Problem. *Mathematics* **2023**, *11*, 4059. [CrossRef]
26. Wang, Z.; Hu, X.; Ma, H.; Xia, W. Learning Multi-Satellite Scheduling Policy with Heterogeneous Graph Neural Network. *Adv. Space Res.* **2024**, *73*, 2921–2938. [CrossRef]
27. Liu, D.; Zhou, G. Deep Reinforcement Learning-Based Attention Decision Network for Agile Earth Observation Satellite Scheduling. *Remote Sens.* **2024**, *16*, 4436. [CrossRef]
28. Zhou, G.; Jin, Z.; Liu, D. Deep Reinforcement Learning-Based Two-Phase Hybrid Optimization for Scheduling Agile Earth Observation Satellites. *Remote Sens.* **2025**, *17*, 1972. [CrossRef]
29. Li, S.; Wang, G.; Chen, J. AEM-D3QN: A Graph-Based Deep Reinforcement Learning Framework for Dynamic Earth Observation Satellite Mission Planning. *Aerospace* **2025**, *12*, 420. [CrossRef]
30. Li, P.; Cui, P.; Wang, H. Mission Sequence Model and Deep Reinforcement Learning-Based Replanning Method for Multi-Satellite Observation. *Sensors* **2025**, *25*, 1707. [CrossRef]
31. Wang, M.; Zhou, Z.; Chang, Z.; Chen, E.; Li, R. Deep Reinforcement Learning for Agile Earth Observation Satellites Scheduling Problem with Variable Image Duration. *Appl. Soft Comput.* **2025**, *169*, 112575. [CrossRef]

32. Feng, X.; Li, Y.; Xu, M. Multi-Satellite Cooperative Scheduling Method for Large-Scale Tasks Based on Hybrid Graph Neural Network and Metaheuristic Algorithm. *Adv. Eng. Inform.* **2024**, *60*, 102362. [CrossRef]
33. Qiu, S.; Tang, H.Q.; Xu, L.W.; Cao, X.B. A Deep Learning-Based Task Denoising Method for Large-Scale Remote Sensing Constellation Mission Planning. *J. Astronaut.* **2026**, *47*, 1057–1070. (In Chinese) [CrossRef]
34. Zhang, R.C.; Ding, G.S.; Duan, C.; Xia, W. A Two-Phase Approach for Agile Earth Observation Satellite Scheduling Based on Deep Reinforcement Learning. *Fire Control Command Control* **2025**, *50*, 63–71. (In Chinese) [CrossRef]
35. Vallado, D.A.; Crawford, P.; Hujsak, R.; Kelso, T.S. Revisiting Spacetrack Report #3. In Proceedings of the AIAA/AAS Astrodynamics Specialist Conference, Keystone, CO, USA, 21–24 August 2006. Paper AIAA 2006-6753. [CrossRef]
36. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017. [CrossRef]
37. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; van den Berg, R.; Titov, I.; Welling, M. Modeling Relational Data with Graph Convolutional Networks. In *The Semantic Web—ESWC 2018; Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2018; Volume 10843, pp. 593–607. [CrossRef]
38. Fey, M.; Lenssen, J.E. Fast Graph Representation Learning with PyTorch Geometric. In Proceedings of the ICLR Workshop on Representation Learning on Graphs and Manifolds, New Orleans, LA, USA, 6 May 2019. [CrossRef]
39. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347. [CrossRef]
40. Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.I.; Abbeel, P. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In Proceedings of the 4th International Conference on Learning Representations, San Juan, Puerto Rico, 2–4 May 2016. [CrossRef]
41. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018; 552p, ISBN 978-0-262-03924-6.
42. Henderson, P.; Islam, R.; Bachman, P.; Pineau, J.; Precup, D.; Meger, D. Deep Reinforcement Learning That Matters. *Proc. AAAI Conf. Artif. Intell.* **2018**, *32*, 3207–3214. [CrossRef]
43. Agarwal, R.; Schwarzer, M.; Castro, P.S.; Courville, A.; Bellemare, M.G. Deep Reinforcement Learning at the Edge of the Statistical Precipice. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 29304–29320.
44. Lv, J.; Wang, C.; Xie, L. Adaptive Distributed Observer Design for Nonlinear Multiagent Systems. *Automatica* **2026**, *183*, 112625. [CrossRef]
45. Yang, G.; Li, H.; Zhao, X.; Yang, J.; Guo, H.; Wei, D.; Huo, L. Optimized Allocation of Irrigation Water Resources Based on Uncertainty: Model Construction and Dynamic Regulation Mechanism. *Water* **2026**, *18*, 612. [CrossRef]

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