



Article

GRIP-MAPPO: Feasibility-Preserving Graph-Relational Insertion-Preference Learning for Downlink-Constrained LEO Remote-Sensing Scheduling

Fangrong Zhou ¹, Guofang Wang ¹, Gang Wen ^{1,*}, Taoyang Wang ², Xin Li ³ , Zonghan Jiao ¹, Jingrun Zhang ¹, Shuangde Huang ⁴ and Jianwen Sun ¹

¹ Electric Power Research Institute, Yunnan Power Grid Co., Ltd., Kunming 650217, China;

zhoufangrong@dlyjy.yn.csg.cn (F.Z.); wangguofang@dlyjy.yn.csg.cn (G.W.);

jiaozonghan@dlyjy.yn.csg.cn (Z.J.); zhangjinrun@yn.csg.cn (J.Z.); sunjianwen@yn.csg.cn (J.S.)

² School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China;

wangtaoyang@whu.edu.cn

³ School of Automation and Electronic Information, Xiangtan University, Xiangtan 411105, China;

lixin_1995@xtu.edu.cn

⁴ Kunming Power Supply Bureau, Yunnan Power Grid Co., Ltd., Kunming 650011, China;

huangshuande@km.yn.csg.cn

* Correspondence: wengang@dlyjy.yn.csg.cn

Highlights

What are the main findings?

- A feasibility-preserving GRIP-MAPPO framework is developed for downlink-constrained LEO remote-sensing scheduling.
- Typed satellite–task–ground relations and bounded insertion preferences are integrated to guide observation insertion and downlink repair.

What are the implications of the main findings?

- In controlled benchmark simulations, GRIP-MAPPO improves observation completion and transmitted-data fraction over greedy, heuristic, and MAPPO-based baselines.
- The framework demonstrates a scalable online scheduling approach for simulated large LEO constellations under downlink bottlenecks and resource constraints.

Abstract

Online scheduling for low-Earth-orbit (LEO) remote-sensing constellations must coordinate short observation windows, sequence-dependent slews, finite onboard resources, intermittent downlink contacts, and rolling command commitment. We study an acquisition-driven downlink-constrained observation problem in which retained observations carry the primary mission value, while their generated data must remain storable until transmitted. This formulation targets observation-centric missions where downlink supports storage feasibility and future acquisition capability; missions with explicit delivery-value or latency requirements can include downlink completion and delivery delay directly in the objective. The central difficulty is suffix-dependent feasibility: a single locally attractive insertion can cascade into downstream infeasibility through altered slew margins, resource levels, and downlink opportunities. We propose Graph-Relational Insertion-Preference Multi-Agent Proximal Policy Optimization (GRIP-MAPPO), a feasibility-preserving learning-to-dispatch framework comprising a relation-aware satellite–task–ground encoder, orbital-group actors that output bounded insertion-preference vectors, and a deterministic scheduler that enforces hard constraints via insertion, repair, and rollback. Thus, GRIP-MAPPO learns dispatch preferences rather than direct task, satellite, or window assignments, while hard-constraint satisfaction remains in an auditable scheduler interface. Controlled benchmark



Academic Editor: Shuanggen Jin

Received: 8 May 2026

Revised: 14 June 2026

Accepted: 21 June 2026

Published: 30 June 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

simulations over 24–288 satellites and six operating regimes show that GRIP-MAPPO improves observation completion and transmitted-data fraction over greedy, heuristic, and graph-based online baselines in the simulated setting, while retaining sub-second replanning latency at reference scale.

Keywords: Low Earth orbit; remote sensing; transmission-aware scheduling; satellite constellation scheduling; multi-agent reinforcement learning; graph neural networks; rolling-horizon planning

1. Introduction

Low-Earth-orbit (LEO) remote-sensing constellations increasingly support time-sensitive Earth-observation services. Their schedulers must replan online because requests arrive continuously, observation time windows (OTWs) are short, onboard resources evolve after every operation, ground-station transmission time windows (TTWs) are intermittent, and only a near-term segment of each rolling plan can be committed. A feasible command sequence must therefore determine not only the observed target, but also the satellite, start time, predecessor–successor attitude transition, energy and storage trajectory, downlink use, and locked execution segment. These coupled timing, attitude, and resource constraints remain central challenges in agile Earth-observation satellite scheduling [1–4].

This paper focuses on acquisition-driven downlink-constrained observation scheduling. An OTW is a time interval in which a satellite can image a target, and a TTW is a ground-station contact interval in which onboard data can be transmitted [5]. In the mission setting considered here, the primary scheduling value is assigned to retained observations: a task is completed when the required observations are included in the feasible rolling schedule. Downlink is modeled as an enabling operation that preserves onboard storage feasibility and supports future acquisition opportunities. Accordingly, each retained observation creates an onboard data item whose storage, transmission, and residual backlog status are tracked. Transmitted-data fraction (TDF) is reported as a transmission-status metric, and age-weighted residual backlog is included as an operational risk term. For delivery-driven missions, such as near-real-time disaster response, tactical surveillance, direct-to-user delivery, or service-level-agreement-based data products, successful downlink and delivery latency can be promoted to direct objective terms.

The main technical difficulty is suffix-dependent feasibility. A candidate observation may satisfy its OTW and deadline locally, but still become unacceptable after insertion because it changes the slew time to the next operation, pushes a later energy state below the safe floor, fills storage before the next usable TTW, creates duplicate coverage, or conflicts with commands already locked in the execution segment. These effects are downstream sequence properties rather than one-step satellite–task masks. A direct neural assignment policy may therefore produce many actions that require rejection or repair, while a purely optimization-based method may become too expensive when satellites, tasks, OTWs, TTWs, ground-station channels, and insertion positions must all be reconsidered at each epoch.

Recent deep reinforcement learning methods have improved the adaptivity of agile Earth-observation scheduling, including deep reinforcement learning (DRL)-based schedule revision, attention-based decision networks, and hybrid DRL optimization. However, directly learned task, satellite, or window decisions remain difficult to certify when downstream sequence feasibility depends on slews, storage, downlink opportunities, and locked rolling segments [6,7]. Graph-relational insertion-preference multi-agent proximal policy optimization (GRIP-MAPPO) is designed around a narrow and auditable scheduler inter-

face. It uses a typed satellite–task–ground graph to summarize resources, active requests, OTW alternatives, task conflicts, satellite coordination, and TTW/downlink opportunities. Orbital-group actors output bounded insertion preferences over profit, urgency, imaging quality, energy cost, memory/downlink pressure, load balance, acceptance threshold, and resource reservation. A deterministic scheduler uses these preferences to rank candidates, insert observations, repair storage violations with feasible downlinks, remove surplus coverage, re-simulate affected suffixes, and commit only the executable rolling segment. The policy can change which candidates are attempted and with what margins, but it cannot bypass timing, resource, downlink, duplicate-coverage, or rolling-commitment checks.

GRIP-MAPPO therefore adopts an auditable learning-to-dispatch interface for downlink-constrained LEO scheduling. The neural policy does not directly generate low-level satellite commands or certify final schedule feasibility; instead, it compresses the high-dimensional graph-relational state into a compact bounded preference vector with physically interpretable dimensions, such as profit, urgency, imaging quality, memory/downlink pressure, and resource reservation. The deterministic scheduler then converts this preference into executable operation sequences through constraint checking, insertion, repair, rollback, and final audit. In this way, learning improves candidate ordering and resource-aware schedule quality, while hard feasibility remains enforced by deterministic scheduling logic.

The main contributions are as follows:

- We formulate a downlink-constrained rolling-horizon LEO remote-sensing scheduling problem with OTW/TTW availability, coverage requirements, sequence-dependent attitude transitions, energy/storage propagation, observation-data accounting, storage-relief downlink, ground-station channel contention, and locked-segment consistency.
- We propose GRIP-MAPPO, a graph-relational insertion-preference framework in which orbital-group MAPPO actors generate a compact bounded preference vector from a typed satellite–task–ground graph, while hard feasibility certification remains outside the neural network.
- We develop a deterministic feasibility-preserving scheduler that converts learned preferences into validated operation sequences through observation insertion, bounded downlink repair, duplicate resolution, suffix resource propagation, rollback, and final C1–C10 audit. We formalize the corresponding scheduler-interface feasibility invariant and evaluate the method through reference-scale comparison, robustness, scalability, transmission-stress sensitivity, ablation analysis, and reduced-scale calibration against exact or bounded references.

2. Related Work

2.1. Optimization and Integrated Observation–Downlink Scheduling

Earth-observation satellite scheduling has long been studied through mathematical programming, constraint programming, graph search, and tailored meta-heuristics. Foundational single- and agile-satellite formulations model OTWs, attitude-transition times, target priorities, and resource constraints explicitly [8–10], while constellation-level extensions add multi-satellite coupling, resource propagation, and sequence-dependent transitions [11,12]. Recent surveys emphasize the same trade-off that motivates this work: explicit constraint modeling is essential, but online scalability remains difficult [1,2].

Exact and decomposition-based methods provide strong references. Mixed-integer programming has been used for multi-satellite and cloud-aware constellation planning [13,14]; exact algorithms, parallel-machine abstractions, branch-cut-and-price, and branch-and-price have been developed for agile or energy-aware observation scheduling [15–18]. Heuristics, neighborhood search, and distributed negotiation protocols trade optimality for

speed and have been used for robustness, data-driven genetic search, large-neighborhood improvement, and dynamic data-relay satellite rescheduling [19–22]. These approaches are important baselines, but their online cost or tuning burden grows when many OTWs, TTWs, channel choices, and sequence positions interact at each rolling epoch. Table 1 summarizes the methodological positioning of GRIP-MAPPO.

Table 1. Position of GRIP-MAPPO relative to representative solution lines. The table compares decision interfaces and feasibility roles rather than assigning checklist scores to broad method families.

Solution Line	Typical Decision Interface	Downlink/Data Role	Feasibility Mechanism	Gap Addressed by GRIP-MAPPO
Exact, CP, or MIP scheduling	Binary or sequence variables over tasks, windows, resources, and sometimes contacts	Explicit when included in the formulation	Solver-enforced constraints and bounds	Strong modeling reference, but repeated online solves can become expensive as OTWs, TTWs, channels, and insertion positions row.
Constructive heuristics and neighborhood search	Handcrafted priority rules, insertion moves, destroy–repair, or local improvement	Usually encoded through rules, penalties, or repair operators	Deterministic checks after insertion or neighborhood moves	Fast and interpretable, but priority surfaces are fixed or require long search under changing urgency and downlink pressure.
Integrated observation–downlink RL	Direct choices over tasks, satellites, windows, or communication actions	Often part of an integrated delivery objective	Masks, penalties, repair, or environment rejection	Adaptive, but direct actions are hard to certify because one choice can change downstream slews, resources, and TTW opportunities.
Graph-based satellite scheduling policies	Graph embeddings used to score tasks, satellites, or construction steps	Problem-dependent; downlink may be absent or simplified	Local masks or downstream repair depending on the implementation	Relational state is useful, but final assignment is still not automatically a sequence-level feasibility certificate.
GRIP-MAPPO	Bounded graph-relational insertion-preference vector consumed by a deterministic scheduler	Storage-relief downlink with TDF and backlog diagnostics computed from the returned sequence	Validated insertion, bounded downlink repair, duplicate resolution, suffix propagation, rollback, and final C1–C10 audit	Adapts the priority surface while keeping hard timing, resource, downlink, and rolling-commitment constraints outside the neural actor.

Transmission-aware work is closest to the present problem. Joint observation–transmission models encode shared storage and ground-station conflicts [23], and integrated imaging–downloading scheduling has been addressed by branch-and-price, genetic algorithms, and more recent agile-satellite formulations [24–27]. Learning-based integrated schedulers include MILP-based imitation learning, hybrid actor–critic methods, hierarchical DRL, and PPO-based auction mechanisms for multi-satellite observation and data transmission or inter-layer link scheduling [28–31]. GRIP-MAPPO differs in objective and interface: observation completion remains the value-bearing outcome, downlink is modeled as constrained storage relief, and TDF/backlog is a deterministic diagnostic of the validated sequence rather than a separate communication-service reward.

2.2. Reinforcement Learning and Graph-Based Satellite Scheduling

Reinforcement learning has been used to replace fixed priority rules with adaptive policies. Neural combinatorial optimization shows that attention-based policies can learn routing and sequencing heuristics [32,33]. In satellite scheduling, MDP formulations, parameter-transfer DRL, distributed multi-agent DRL, high–low-orbit mission planning, periodic agile-constellation scheduling, satellite range scheduling, and varying-scale generalization have all been explored [34–40]. These methods demonstrate the value of adaptation, but many policies directly select tasks, satellites, or windows. Such actions are difficult

to certify in large constellations because feasibility depends on the downstream sequence, not only on the selected local pair.

Multi-agent RL is natural for constellations because satellites interact through shared targets, overlapping OTWs, and ground-station contention. Centralized training with decentralized execution has been effective in cooperative domains [41,42]; however, assigning one agent to every satellite can increase non-stationarity, whereas a single centralized actor scales poorly. GRIP-MAPPO uses orbital groups as agents, giving each actor a localized coordination role while sharing a graph encoder and centralized critic.

Graph learning is also well suited to this domain. Satellites, tasks, ground stations, OTWs, conflicts, and TTWs form a typed relational state; heterogeneous attention provides a general representation mechanism [43]. Satellite-specific graph-learning studies have learned multi-satellite scheduling policies or neural construction rules for time-dependent transition times [44,45]. GRIP-MAPPO uses this relational representation not to output a final assignment, but to parameterize a bounded insertion-preference vector that is consumed by an explicit scheduler.

2.3. Learning-Guided Dispatching and Feasibility Interfaces

The proposed method is related to learning-guided optimization, where neural models accelerate or steer structured combinatorial procedures. Learning-to-branch methods guide mixed-integer programming search [46,47]; neural large-neighborhood search learns destroy, repair, or selection policies within a classical search loop [48]; and learning-to-dispatch estimates priority parameters while deterministic rules construct schedules [49,50]. These ideas share the principle of using learning for guidance rather than replacing the combinatorial procedure.

GRIP-MAPPO follows this principle but uses a stricter feasibility interface. Hybrid MIP or neural LNS methods may explore infeasible partial states and rely on the solver or repair loop to recover feasibility. GRIP-MAPPO instead restricts the learned action to low-dimensional insertion preferences and lets the deterministic scheduler perform only validated insertions, repairs, rollbacks, suffix propagation, and final audit. The target difficulty is therefore suffix-dependent feasibility in a coupled satellite–task–ground system, rather than only priority prediction for a machine queue, iterative destroy-and-repair search, or direct neural assignment over a large action space.

The gap left by these lines of work is not the absence of graph learning, MAPPO, or insertion heuristics individually. It is the lack of a compact neural action interface that can adapt to a typed satellite–task–ground state while preserving suffix-level feasibility through deterministic construction and audit.

3. Problem Formulation and Operation Model

Figure 1 shows the physical entities in the mission scene, while this section defines the temporal, resource, and transmission-status semantics used by the scheduler. At each rolling epoch, the scheduler receives active tasks, satellite resource states, candidate observation time windows (OTWs), transmission time windows (TTWs), transition-time functions, link-rate functions, inherited locked operations, and the onboard observation-data state induced by previous observations. The learned policy does not directly output task assignments, start times, or ground-station TTW choices. It outputs bounded scheduling preferences; a deterministic scheduler converts those preferences into operation sequences and accepts only schedules that satisfy the feasibility checks summarized below.

The formulation is downlink-constrained rather than a full communication-service objective. Observation tasks determine the primary observation-completion value, but every accepted observation produces data that must remain stored until a valid downlink

removes it. The model therefore includes observation-data-item backlog and age-weighted untransmitted-data risk. These additions make downlink pressure observable and measurable without changing the method: the same scheduler operations determine observation completion, storage feasibility, and transmission-status metrics.

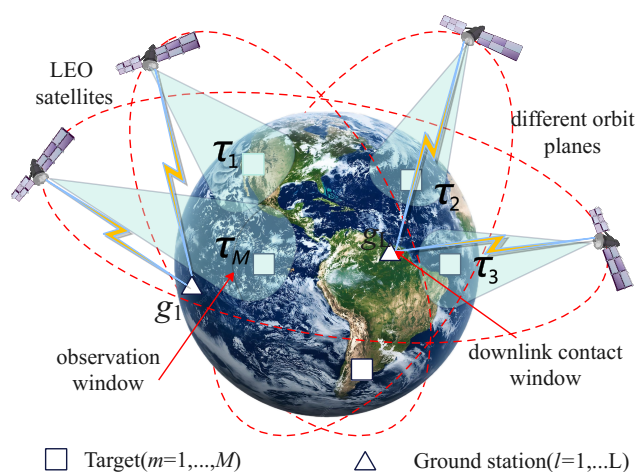


Figure 1. Physical mission scene for downlink-constrained LEO remote-sensing constellation scheduling. Satellites acquire targets during observation time windows (OTWs) and offload generated data during intermittent transmission time windows (TTWs) with ground stations.

The formulation separates hard operational constraints from implementation-specific fidelity. OTWs, TTWs, attitude-transition times, link rates, and resource updates may be generated by simplified simulators, high-fidelity mission analysis tools, or flight-dynamics software. The scheduler only requires deterministic interfaces for these quantities during a rolling epoch. This separation keeps the proposed learning interface independent of a particular orbit propagator or communication-link model, while clarifying that the experimental simulator is used for controlled algorithmic comparison rather than flight qualification.

3.1. Rolling State and Operation Primitives

At epoch t , the look-ahead horizon and committed execution segment are

$$H_t = [t, t + \Delta H], \quad H_t^{\text{exe}} = [t, t + \Delta h], \quad 0 < \Delta h < \Delta H. \quad (1)$$

Operations whose start times lie in H_t^{exe} are locked after the epoch; later operations remain tentative and may be revised at the next epoch. The satellite, ground-station, and active-task sets are

$$\begin{aligned} \mathcal{S} &= \{s_1, \dots, s_N\}, \\ \mathcal{G} &= \{g_1, \dots, g_L\}, \\ \mathcal{T}_t &= \{\tau_1, \dots, \tau_{M(t)}\}. \end{aligned} \quad (2)$$

Each task is represented by

$$\tau_j = (\ell_j, p_j, a_j, d_j, \delta_j, m_j, q_j, D_j, \kappa_j), \quad (3)$$

where the entries denote target location, profit, arrival time, deadline, imaging duration, required number of observations, imaging mode, generated data volume, and urgent-task flag. An observation time window (OTW) and a transmission time window (TTW) are denoted by

$$w_{ijr}^{\text{obs}} = [t_{ijr}^-, t_{ijr}^+], \quad w_{ilr}^{\text{dl}} = [t_{ilr}^{\text{dl},-}, t_{ilr}^{\text{dl},+}]. \quad (4)$$

A satellite operation is written as

$$o = (\chi, \eta, t_s, t_e, \theta_i, d_i^{\text{tx}}), \quad (5)$$

where χ is the operation type, η is a task or ground-station identifier, t_s and t_e are start and end times, θ_i is the pointing direction, and d_i^{tx} is the downlinked volume for a downlink operation. A rolling schedule $\Pi_t = \{\pi_i^t\}_{i=1}^N$ is a set of time-ordered operation sequences, one sequence for each satellite, together with inherited locked operations.

The notation separates requests, time-window opportunities, and executed operations. A task stores value, timing, mode, data volume, coverage, and urgency information; OTWs and TTWs describe the time intervals in which service can occur; and each satellite sequence contains observation, downlink, and idle operations ordered by start time. Energy $E_i(t)$, storage $B_i(t)$, attitude direction θ_i , and transition time T_i^{slew} are propagated whenever an operation is inserted or removed. This separation is used throughout the scheduler: learned preferences rank candidates, while feasibility is decided only after suffix propagation.

The simulator propagates each accepted sequence by applying operation-specific update functions to energy and storage. For a candidate operation o' inserted after predecessor o , the affected suffix begins with

$$(E_i^+, B_i^+) = \mathcal{P}_i(o' | o, E_i^-, B_i^-), \quad (6)$$

where E_i^- and B_i^- are the propagated resource states before o' . For an observation of task τ_j , the simplified accounting used by the scheduler can be summarized as

$$E_i(t_e(o')) = E_i(t_s(o')) - e_i^{\text{slew}}(o, o') - e_i^{\text{img}}(q_j, \delta_j), \quad (7)$$

$$B_i(t_e(o')) = B_i(t_s(o')) + D_j. \quad (8)$$

For a downlink starting at v with duration Δv inside TTW w_{ilr}^{dl} , the transmitted volume must satisfy

$$0 \leq d_i^{\text{tx}} \leq \min \left\{ B_i(v), \int_v^{v+\Delta v} R_{il}(\xi) d\xi \right\}. \quad (9)$$

The detailed orbit, link, and attitude implementation can be replaced by higher-fidelity models as long as they provide OTWs, TTWs, transition times, and deterministic resource propagation to the scheduler.

3.2. Observation-Data Accounting and Transmission-Status Metrics

To strengthen the downlink model without changing the scheduler interface, each accepted observation creates an observation-data item

$$z = (j, i, t_z^{\text{obs}}, D_z, \kappa_z), \quad D_z = D_j, \quad \kappa_z = \kappa_j, \quad (10)$$

where j is the parent task, i is the producing satellite, t_z^{obs} is the observation end time, D_z is the generated or residual volume, and κ_z inherits the urgency flag. Let $\mathcal{Z}_i(\xi)$ be the multiset of untransmitted observation-data items stored on satellite s_i at time ξ . Observation and downlink operations update this set consistently with the scalar storage state:

$$\mathcal{Z}_i(t_e(o^{\text{obs}})) = \mathcal{Z}_i(t_s(o^{\text{obs}})) \cup \{z\}, \quad (11)$$

$$\sum_{z \in \mathcal{Z}_i(t_e(o^{\text{dl}}))} D_z = \sum_{z \in \mathcal{Z}_i(t_s(o^{\text{dl}}))} D_z - d_i^{\text{tx}}. \quad (12)$$

When a downlink transmits only part of the onboard backlog, the transmission-status evaluator uses a fixed attribution rule: urgent data items are served first in increasing

observation end time, non-urgent items are then served in increasing observation end time, and remaining ties are broken by task and satellite identifiers. A partially served item keeps a residual volume in $\mathcal{Z}_i$. This rule does not introduce a scheduling decision; it only attributes transmitted volume to observation-data items for reproducible reporting and risk evaluation.

For task τ_j , the generated observation-data volume, transmitted-data fraction, and untransmitted-data backlog within the horizon are

$$\begin{aligned} D_j^{\text{gen},H} &= n_j D_j, \\ \delta_j^{\text{tr},H} &= \min \left\{ 1, \frac{D_j^{\text{tx},H}}{D_j^{\text{gen},H} + \epsilon_D} \right\}, \quad D_j^{\text{gen},H} > 0, \\ U_j^H &= \left[D_j^{\text{gen},H} - D_j^{\text{tx},H} \right]_+. \end{aligned} \quad (13)$$

where $D_j^{\text{tx},H}$ is the volume attributed to observation-data items of task j that has been transmitted by the end of the look-ahead horizon. Aggregate TDF is computed over tasks with generated observation data; tasks for which the returned schedule generates no data are omitted from the aggregate transmission-status score. The primary completion indicator c_j remains observation-based; $\delta_j^{\text{tr},H}$ and U_j^H are transmission-status quantities. They let the evaluation distinguish retained observation data that remain onboard from data that are relieved through available TTWs, while preserving the same actor output and scheduler operations.

3.3. Objective and Scheduler Interface

The number of retained observations for task τ_j and its observation-completion indicator are

$$c_j = \mathbb{I}(n_j \geq m_j), \quad n_j = \sum_i \sum_r x_{ijr}, \quad (14)$$

where $x_{ijr} = 1$ if s_i observes τ_j in OTW r . The indicator c_j is intentionally tied to retained observations rather than to a separate communication-service profit. Downlink affects the objective through storage feasibility, storage evolution, and the age-weighted risk of residual untransmitted observation data. This choice matches observation-driven missions in which the scheduler must acquire high-value imagery while preventing onboard data from blocking future acquisitions.

The rolling-horizon objective is

$$\max_{\Pi_t} J_{\text{roll}}(\Pi_t) = \sum_{\tau_j \in T_t} \bar{p}_j \left(1 + \lambda_u \kappa_j e^{-\Delta_j^{\text{obs}}/\tau_u} \right) c_j - \lambda_{\text{op}} \Phi(\Pi_t), \quad (15)$$

where $\bar{p}_j$ is normalized profit, Δ_j^{obs} is urgent observation-response lag, λ_u controls urgent-task emphasis, and λ_{op} controls the price of conservative operation. The operation cost is

$$\Phi(\Pi_t) = \Psi_{\text{res}}(\Pi_t) + \Psi_{\text{load}}(\Pi_t) + \Psi_{\text{rev}}(\Pi_t). \quad (16)$$

The three components are normalized before combination. The learning architecture and action dimension are unchanged by this objective: every term in Φ is computed after the deterministic scheduler returns Π_t .

The strengthened resource-and-untransmitted-data risk component is

$$\begin{aligned}\Psi_{\text{res}} &= \frac{1}{N} \sum_{i=1}^N (R_i^E + R_i^B + R_i^D), \\ R_i^E &= \alpha_E \left[\frac{E_i^{\text{safe}} - E_i^{\text{min},H}}{E_i^{\text{cap}}} \right]_+, \\ R_i^B &= \alpha_B \left[\frac{B_i^{\text{max},H} - B_i^{\text{safe}}}{B_i^{\text{cap}}} \right]_+, \\ R_i^D &= \alpha_D \frac{1}{B_i^{\text{cap}}} \sum_{z \in \mathcal{Z}_i(t+\Delta H)} D_z \left(1 + \alpha_A a_z^H + \alpha_U \kappa_z \right),\end{aligned}\quad (17)$$

where $E_i^{\text{min},H}$ is the minimum propagated energy in the horizon, $B_i^{\text{max},H}$ is the maximum propagated storage occupancy, $a_z^H = (t + \Delta H - t_z^{\text{obs}}) / (\Delta H + \epsilon_A)$ is the normalized age of an untransmitted observation-data item, and κ_z is its urgency flag. The term R_i^D refines the simpler undownloaded-volume penalty by distinguishing recent low-urgency backlog from older or urgent onboard data. It does not create a new downlink-profit task class; it penalizes the returned operation sequence when it leaves risky data onboard.

The load-balance term is based on Jain's index,

$$J_t = \begin{cases} 1, & \sum_i L_i = 0, \\ \frac{(\sum_i L_i)^2}{N \sum_i L_i^2 + \epsilon_J}, & \sum_i L_i > 0, \end{cases} \quad \Psi_{\text{load}} = 1 - J_t. \quad (18)$$

The zero-load case is defined as perfectly balanced. The revision term is

$$\Psi_{\text{rev}} = \frac{|\Pi_t^{\text{tent}} \Delta \Pi_{t-1}^{\text{tent}}|}{|\Pi_t^{\text{tent}} \cup \Pi_{t-1}^{\text{tent}}| + \epsilon_R}, \quad (19)$$

where Π_t^{tent} is the non-committed part of the look-ahead schedule.

For a fixed rolling state and inherited committed schedule, define

$$\begin{aligned}\mathcal{F}_t(\Pi^{\text{lock}}) &= \{\Pi_t : \Pi_t \text{ contains all locked operations in } \Pi^{\text{lock}}, \\ &\quad \Pi_t \text{ satisfies constraints C1–C10 over } H_t\}.\end{aligned}\quad (20)$$

Definition 1 (suffix-dependent feasibility). *Given a rolling schedule Π_t , inserting a candidate observation c at position h is locally admissible if it satisfies its own OTW, deadline, sensor-mode, and immediate predecessor–successor slew checks. Let $\text{Suf}(c, h)$ denote the inserted operation and all subsequent operations whose timing or resource states may be changed by this insertion. The insertion is suffix-feasible only if, after re-propagating $\text{Suf}(c, h)$, the resulting schedule belongs to $\mathcal{F}_t(\Pi^{\text{lock}})$. Thus, local admissibility does not imply suffix feasibility:*

$$\text{Loc}_t(c, h; \Pi_t) = 1; \nRightarrow \Pi_t \oplus_h c \in \mathcal{F}_t(\Pi^{\text{lock}}). \quad (21)$$

The ideal rolling problem is

$$\Pi_t^* \in \arg \max_{\Pi_t \in \mathcal{F}_t(\Pi^{\text{lock}})} J_{\text{roll}}(\Pi_t). \quad (22)$$

GRIP-MAPPO does not solve (22) exactly. Given graph state s_t and bounded preference action a_t , the deterministic scheduler returns

$$\Pi_t(a_t) = \mathfrak{S}(s_t, a_t) \in \mathcal{F}_t(\Pi^{\text{lock}}). \quad (23)$$

The policy therefore controls the priority surface, acceptance threshold, and resource-reservation margin, while feasibility and transmission-status accounting are supplied by deterministic sequence construction and validation.

3.4. Feasibility Constraints and Markov Game

The deterministic scheduler validates each candidate against ten constraint groups. These constraints define the feasible set used by the scheduler and are enforced after every accepted insertion, repair, deletion, and merge operation. For an accepted observation candidate, the OTW, deadline, and sensor-mode checks (C1–C3) are

$$x_{ijr} = 1 \Rightarrow t_{ijr}^- \leq u_{ijr} \leq t_{ijr}^+ - \delta_j, \quad (24)$$

$$x_{ijr} = 1 \Rightarrow u_{ijr} + \delta_j \leq d_j, \quad (25)$$

$$x_{ijr} = 1 \Rightarrow q_j \in Q_i. \quad (26)$$

The coverage and duplicate-control checks (C4–C5) are

$$n_j = \sum_i \sum_r x_{ijr}, \quad c_j = \mathbb{I}(n_j \geq m_j), \quad (27)$$

$$0 \leq n_j \leq m_j. \quad (28)$$

The last inequality is imposed after duplicate resolution. A surplus observation does not increase task value once the required coverage has been met, but it still consumes imaging time, attitude maneuver margin, storage, energy, and downlink capacity. Therefore, if a candidate construction temporarily creates $n_j > m_j$, the scheduler removes surplus observations by score order and re-simulates the affected suffixes before returning the schedule. To avoid counting repeated coverage from a single satellite within one horizon, the scheduler also enforces

$$\sum_r x_{ijr} \leq 1, \quad \forall i, j. \quad (29)$$

For any two consecutive operations o and o' on satellite s_i , the attitude-transition check (C6) requires

$$t_s(o') \geq t_e(o) + T_i^{\text{slew}}(o, o'). \quad (30)$$

Energy and storage feasibility (C7–C8) is evaluated at all operation events and simulator check points:

$$E_i(\xi) \geq E_i^{\min}, \quad 0 \leq B_i(\xi) \leq B_i^{\text{cap}}, \quad \forall \xi \in \Xi_i(\Pi_t), \quad (31)$$

where $\Xi_i(\Pi_t)$ is the set of operation starts, operation ends, and intermediate check points used by the simulator. For a downlink assigned to TTW (i, l, r) , the downlink feasibility check (C9) requires

$$t_{ilr}^{\text{dl},-} \leq v_{ilr}, \quad v_{ilr} + \Delta v_{ilr} \leq t_{ilr}^{\text{dl},+}, \quad (32)$$

$$0 \leq d_{ilr}^{\text{tx}} \leq \min \left\{ B_i(v_{ilr}), \int_{v_{ilr}}^{v_{ilr} + \Delta v_{ilr}} R_{il}(\xi) d\xi \right\}. \quad (33)$$

The finite-channel version of C9 also includes ground-station contention. If station g_l has C_l^{ch} simultaneous channels, overlapping downlinks satisfy

$$\sum_{i,r} \mathbb{I}\{\xi \in [v_{ilr}, v_{ilr} + \Delta v_{ilr}]\} \leq C_l^{\text{ch}}, \quad \forall l, \xi. \quad (34)$$

When an independent-channel abstraction is used, C_l^{ch} is set large enough that this constraint is nonbinding and the assumption is reported with the experimental setting. C9 also enforces data conservation through (9): transmitted volume cannot exceed the data available onboard at the start of the TTW segment. Observation-data-level transmission attribution in (12) is therefore a transmission-status evaluation of the same feasible operation sequence, not an additional scheduling action. Finally, rolling consistency (C10) is enforced by

$$\Pi_t|_{H_t^{\text{exe}}} \supseteq \Pi^{\text{lock}}, \quad \Pi_t \subseteq H_t, \quad (35)$$

and the scheduler is not allowed to shift or remove locked operations.

The learning problem is a cooperative-graph Markov game

$$\mathcal{M} = \langle \mathcal{K}, \mathcal{S}^{\text{env}}, \{\mathcal{O}_k\}_{k=1}^K, \{\mathcal{A}_k\}_{k=1}^K, P, R, \gamma, \mu_0 \rangle. \quad (36)$$

Here $\mathcal{K}$ denotes orbital-group agents; $\mathcal{S}^{\text{env}}$ contains graph state, active tasks, resources, OTWs/TTWs, onboard observation-data state, and locked or tentative schedules; $\mathcal{O}_k$ is the group-local graph readout; and $\mathcal{A}_k = \mathbb{R}^8$ is the raw preference-action space before bounded transformation. The shared reward R is computed from the committed segment after the scheduler has constructed, repaired, validated, and transmission-status-evaluated the schedule. The transition P is induced by the deterministic scheduler, the committed-segment simulator, the observation-data accounting rules, and the exogenous scenario stream. Conditioned on a sampled episode stream, candidate OTWs/TTWs, and task arrivals, the scheduler, simulator, and transmission-status evaluation are deterministic; without this conditioning, P also includes the task-arrival and window-generation processes.

4. Proposed GRIP-MAPPO Method

GRIP-MAPPO is described here at the interface level. The neural component maps the typed scheduling graph to a compact eight-dimensional preference action, and the deterministic scheduler converts this action into a feasible rolling sequence. To reduce implementation-level detail, this section focuses on the graph representation, bounded action interface, and feasibility-preserving scheduler invariant.

4.1. Heterogeneous Satellite–Task–Ground Graph

At epoch t , the environment constructs a typed multigraph

$$G_t = (\mathcal{V}_s, \mathcal{V}_\tau, \mathcal{V}_g, \mathcal{E}_{ss}, \mathcal{E}_{s\tau}, \mathcal{E}_{\tau\tau}, \mathcal{E}_{sg}). \quad (37)$$

Satellite, task, and ground-station nodes represent resources, observation requests, and TTW/downlink opportunities. The four edge types are kept separate because they have different scheduling meanings: satellite–satellite edges encode coordination and coverage overlap, satellite–task multiedges encode OTW alternatives, task–task edges encode competition for shared feasible intervals, and satellite–ground edges encode downlink pressure and storage-relief opportunities.

The graph uses normalized satellite, task, and ground-station features together with typed edge attributes. Satellite–task multiedges retain OTW timing, geometric quality, transition lower bounds, and resource cost; task–task edges summarize competition for shared

feasible intervals; satellite–ground edges encode TTW timing, remaining capacity, link rate, and near-term storage relief; and satellite–satellite edges provide coordination context across neighboring orbital positions. These relation types give the actor the information needed to adapt insertion preferences, while final sequence feasibility remains enforced by the deterministic scheduler rather than by the graph encoder alone. Figure 2 illustrates the resulting heterogeneous scheduling graph.

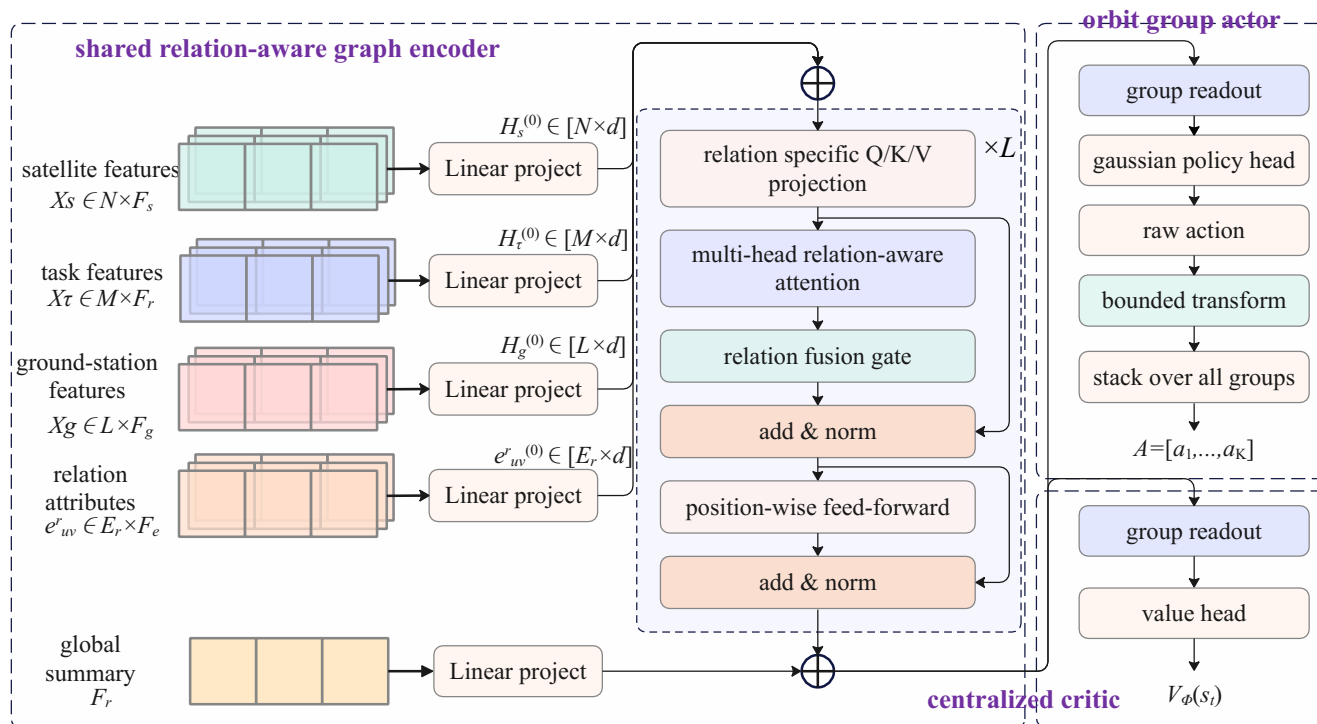


Figure 2. Heterogeneous scheduling graph construction. The physical rolling state is converted into a typed satellite–task–ground multigraph. Node features summarize resources, observation requests, and ground-station TTWs; relation-specific edge attributes encode OTW feasibility, coordination, conflict, and storage-relief TTW/downlink opportunities.

The geometric quality used in edge scoring is

$$Q_{ijr}^{\text{geo}} = \text{norm} \left(\max_{u \in [t_{ijr}^-, t_{ijr}^+ - \delta_j]} \frac{1}{\delta_j} \int_u^{u+\delta_j} \sin(\epsilon_i(\ell_j, \xi)) d\xi \right), \quad (38)$$

where ϵ_i is the elevation angle. Deterministic truncation bounds graph density; it can affect quality if useful candidates are removed, but it cannot create hard-constraint violations because retained candidates still pass downstream validation.

For reproducibility, the retained graph is defined by four truncation budgets rather than by a learned sparsifier: each satellite keeps at most q satellite–neighbor edges, each satellite keeps at most p satellite–task OTW multiedges, each task keeps at most c_τ task–conflict edges, and each satellite keeps at most r_g satellite–ground TTW edges. Ties are broken deterministically by normalized OTW quality, time slack, and identifier order. A candidate removed by truncation is not considered by the scheduler in that epoch, so truncation can reduce objective quality; however, every retained candidate still undergoes the full insertion, resource, downlink, and final committed-segment validation.

4.2. Graph Encoder and Preference Action

A shared relation-aware graph encoder maps G_t to node embeddings. For relation r and neighbor set $\mathcal{N}_r(u)$, the edge-aware attention message is

$$M_r^\ell(u) = \sum_{v \in \mathcal{N}_r(u)} \alpha_{uv}^{r,\ell} (W_V^{r,\ell} h_v^\ell + W_E^{r,\ell} e_{uv}^r), \quad (39)$$

$$\alpha_{uv}^{r,\ell} = \text{softmax}_{v \in \mathcal{N}_r(u)} \frac{(W_Q^{r,\ell} h_u^\ell)^\top (W_K^{r,\ell} h_v^\ell + W_{EK}^{r,\ell} e_{uv}^r)}{\sqrt{d}}, \quad (40)$$

where e_{uv}^r denotes the retained edge attributes. Relation messages are then combined by a node-wise gate,

$$g_{u,r}^\ell = \text{softmax}_{r \in \mathcal{R}_u} (a_r^\top [h_u^\ell \| M_r^\ell(u)]), \quad (41)$$

and the layer update is

$$h_u^{\ell+1} = \text{FFN} \left(\text{LN} \left(h_u^\ell + \sum_{r \in \mathcal{R}_u} g_{u,r}^\ell M_r^\ell(u) \right) \right). \quad (42)$$

This presentation keeps the encoder definition explicit without committing the method to a particular number of heads or hidden width; those are implementation hyperparameters. If a node has no retained neighbor of a given relation, that relation contributes zero message and the residual update preserves its own state. This case is important for satellites or tasks that are temporarily isolated after graph truncation.

Satellites are partitioned into orbital-group agents,

$$\mathcal{S} = \bigcup_{k=1}^K \mathcal{S}_k. \quad (43)$$

The partition groups satellites with close orbital planes or neighboring orbital phases and uses a balancing rule to avoid highly uneven group sizes. It gives each actor a physically meaningful coordination region while keeping the number of agents much smaller than the number of satellites.

The actor observation is a pooled readout,

$$o_k^t = [\text{pool}_{s_i \in \mathcal{S}_k} h_i^s \parallel \text{pool}_{\tau_j \in \mathcal{T}_k} h_j^\tau \parallel \text{pool}_{g_l \in \mathcal{G}_k} h_l^g \parallel \bar{r}_t]. \quad (44)$$

Here $\mathcal{T}_k$ contains active tasks incident to at least one satellite in $\mathcal{S}_k$ through a retained OTW edge, $\mathcal{G}_k$ contains ground stations incident to satellites in the group through retained TTW edges, and $\bar{r}_t$ is a compact normalized summary of global resource pressure, load distribution, and rolling revision statistics shared by all group actors. The pooling operator is permutation-invariant, so the policy does not depend on an arbitrary ordering of satellites, tasks, or stations.

For raw action $\tilde{a}_k \in \mathbb{R}^8$, the first six coordinates become normalized weights,

$$w_m^k = \sigma(\tilde{a}_m^k), \quad \omega_m^k = \frac{w_m^k}{\sum_{r=1}^6 w_r^k + \epsilon_a}, \quad m = 1, \dots, 6. \quad (45)$$

The last two coordinates control candidate acceptance and resource reservation,

$$\beta_k = \beta_{\min} + (\beta_{\max} - \beta_{\min}) \sigma(\tilde{a}_7^k), \quad \rho_k = \rho_{\max} \sigma(\tilde{a}_8^k). \quad (46)$$

Thus,

$$a_k = (\omega_p^k, \omega_d^k, \omega_q^k, \omega_e^k, \omega_b^k, \omega_l^k, \beta_k, \rho_k). \quad (47)$$

The eight bounded coordinates have the following scheduler-level meanings. The first six coordinates are normalized ranking weights: ω_p^k emphasizes task profit $\bar{p}_j$, ω_d^k emphasizes deadline urgency u_j^{urg} , ω_q^k emphasizes geometric imaging quality Q_{ijr}^{geo} , ω_e^k penalizes estimated energy consumption ΔE_{ijr} , ω_b^k penalizes storage and downlink pressure ΔB_{ijr} , and ω_l^k penalizes recently overloaded satellites through $\bar{L}_i$. The remaining two coordinates control deterministic scheduling decisions: β_k is the group-specific acceptance threshold for filtering low-score candidates, and ρ_k is the resource-reservation margin used to tighten suffix-level energy and storage feasibility checks. No additional action coordinate is introduced for transmission-status evaluation, because downlink feasibility is handled by deterministic repair and suffix propagation rather than by a separate neural action dimension.

4.3. Preference-Guided Constrained Scheduler

For candidate (i, j, r) in group k ,

$$\begin{aligned} \text{Score}_{ijr}^k = & \omega_p^k \bar{p}_j + \omega_d^k u_j^{\text{urg}} + \omega_q^k Q_{ijr}^{\text{geo}} \\ & - \omega_e^k \Delta E_{ijr} - \omega_b^k \Delta B_{ijr} - \omega_l^k \bar{L}_i. \end{aligned} \quad (48)$$

All terms in (48) are normalized before scoring. The term u_j^{urg} is a normalized deadline-and-urgency score that increases for urgent tasks and shorter remaining slack; ΔE_{ijr} and ΔB_{ijr} are incremental energy and storage costs estimated for the candidate and its immediate transition context; and $\bar{L}_i$ is the normalized recent load of satellite s_i . The score therefore ranks candidates by learned value, urgency, quality, resource cost, and load balance, but it does not certify feasibility. Candidates below β_k are ignored. For insertion between o^- and o^+ ,

$$u_{\min} = \max\{t_{ijr}^-, t_e(o^-) + T_i^{\text{slew}}(o^-, j)\}, \quad (49)$$

$$u_{\max} = \min\{t_{ijr}^+ - \delta_j, d_j - \delta_j, t_s(o^+) - T_i^{\text{slew}}(j, o^+) - \delta_j\}. \quad (50)$$

The propagated affected suffix, rather than only the immediate successor state, must satisfy the reservation margins

$$E_i^{\min, H}(c) \geq E_i^{\min} + \rho_k E_i^{\text{cap}}, \quad (51)$$

$$B_i^{\max, H}(c) \leq B_i^{\text{cap}}(1 - \rho_k). \quad (52)$$

When primary scores are nearly tied, the scheduler applies the fixed deterministic tie term

$$\begin{aligned} \text{Tie}(c) = & \eta_Q Q_{ijr}^{\text{geo}}(u) + \eta_E \bar{E}_i(c) + \eta_B \bar{B}_i(c) \\ & - \eta_T \frac{T_i^{\text{extra}}(c)}{\Delta H + \epsilon_T} - \eta_R \Delta_{ijr}^{\text{rev}}(c), \end{aligned} \quad (53)$$

where $\bar{E}_i(c)$ and $\bar{B}_i(c)$ denote the normalized post-insertion energy and storage margins obtained after suffix propagation. The fixed η coefficients only break construction ties and are not learned action dimensions.

The scheduler uses the following surrogate only to induce a deterministic ranking among feasible construction choices:

$$\begin{aligned} J_{\text{sch}}(\Pi_t | a_t) = & \sum_{(i,j,r) \in \Pi_t^{\text{obs}}} \text{Score}_{ijr}^{g(i)} + \epsilon_{\text{tie}} \sum_{c \in \Pi_t^{\text{obs}}} \text{Tie}(c) \\ & - C_{\text{rep}}(\Pi_t), \end{aligned} \quad (54)$$

where $g(i)$ is the orbital group of satellite s_i , C_{rep} penalizes deterministic repair or revision choices used only for ordering, and ϵ_{tie} is chosen small enough that ties do not override the primary score in (48). This surrogate is not a second mission objective and is not optimized to global optimality. It defines the candidate order used by the finite construction operator

$$\Pi_t(a_t) = \text{Construct}(s_t, a_t; \text{rank}_{J_{\text{sch}}}), \quad (55)$$

which greedily applies the induced ranking, validates every accepted modification, and returns a feasible schedule through Algorithm 1.

Algorithm 1 Compact preference-guided constrained scheduler

Require: Rolling state s_t , bounded group actions $\{a_k\}_{k=1}^K$, locked schedule Π^{lock}

- 1: Initialize satellite sequences with locked operations.
 - 2: **for** each orbital group k in parallel **do**
 - 3: Rank retained observation candidates by (48) and (53); discard candidates below β_k .
 - 4: **for** each ranked candidate **do**
 - 5: Test insertion bounds (49) and (50), propagate the affected suffix, and check resource margins.
 - 6: If a storage violation appears, apply bounded downlink repair.
 - 7: Accept the candidate only if the applicable C1–C10 checks pass.
 - 8: **end for**
 - 9: **end for**
 - 10: Merge group-local sequences, recompute task coverage, and remove duplicate coverage.
 - 11: Apply one bounded fill pass to still-incomplete tasks.
 - 12: Run the final C1–C10 audit; roll back infeasible modifications if needed.
 - 13: Commit executable operations and return Π_t .
-

Algorithm 1 keeps only the scheduler interface in the main text. Implementation details of the downlink-repair routine are summarized rather than listed as a separate algorithm: repair is triggered only by storage excess, tests retained TTW segments, accepts a downlink only when it reduces storage excess without introducing non-storage violations, and is bounded by R_{max} . The normalized storage excess used by this check is

$$\mathcal{E}_i^B(\pi_i) = \sum_{\xi \in \Xi_i(\pi_i)} \left[\frac{B_i(\xi) - B_i^{\text{cap}}(1 - \rho_k)}{B_i^{\text{cap}} + \epsilon_B} \right]_+. \quad (56)$$

The neural actor only changes candidate priorities, the acceptance threshold, and the reservation margin. Observation insertions, induced downlinks, duplicate removals, fill-pass updates, rollback, and final audit are all re-simulated before commitment; infeasible candidates are rejected rather than corrected by the neural policy itself. Figure 3 summarizes the insertion-and-repair procedure.

4.4. Scheduler-Interface Feasibility Invariant and Compatibility

The scheduler guarantee is an interface invariant. It relies on three conditions: the locked schedule is initially feasible; suffix propagation is sound for the modeled timing, attitude, energy, storage, TTW, channel, and data-conservation checks; and every accepted modification is re-simulated and audited before commitment.

Proposition 1 (scheduler-interface feasibility invariant). *Under these conditions, Algorithm 1 returns either the inherited feasible schedule or a modified schedule $\Pi_t \in \mathcal{F}_t(\Pi^{\text{lock}})$ for any bounded preference action a_t .*

Proof. The scheduler starts from the locked feasible sequence and modifies it only through validated insertions, bounded downlink repairs, duplicate removals, fill-pass updates, and rollback decisions. Each accepted modification is followed by suffix propagation and the applicable C1–C10 checks. Rejected candidates leave the current feasible sequence unchanged, and the final global audit rolls back infeasible merged modifications. By induction over accepted modifications, the returned sequence preserves the locked operations and satisfies C1–C10. $\square$

Corollary 1 (policy cannot directly violate hard constraints). *The neural actor changes the candidate ordering, acceptance threshold, and reservation margin only. Since every accepted modification is checked by Algorithm 1, any bounded MAPPO action either yields a feasible sequence or causes infeasible candidates to be rejected.*

Corollary 2 (transmission-status compatibility). *The observation-data variables in (10)–(13) are deterministic functions of accepted observation and downlink operations. They refine scoring and diagnostics without adding another neural action coordinate or operation type.*

Thus, Proposition 1 is an interface-safety claim, not an optimality or completeness claim. Candidate truncation, greedy ordering, and bounded repair may still exclude opportunities that a longer exact or neighborhood search could exploit.

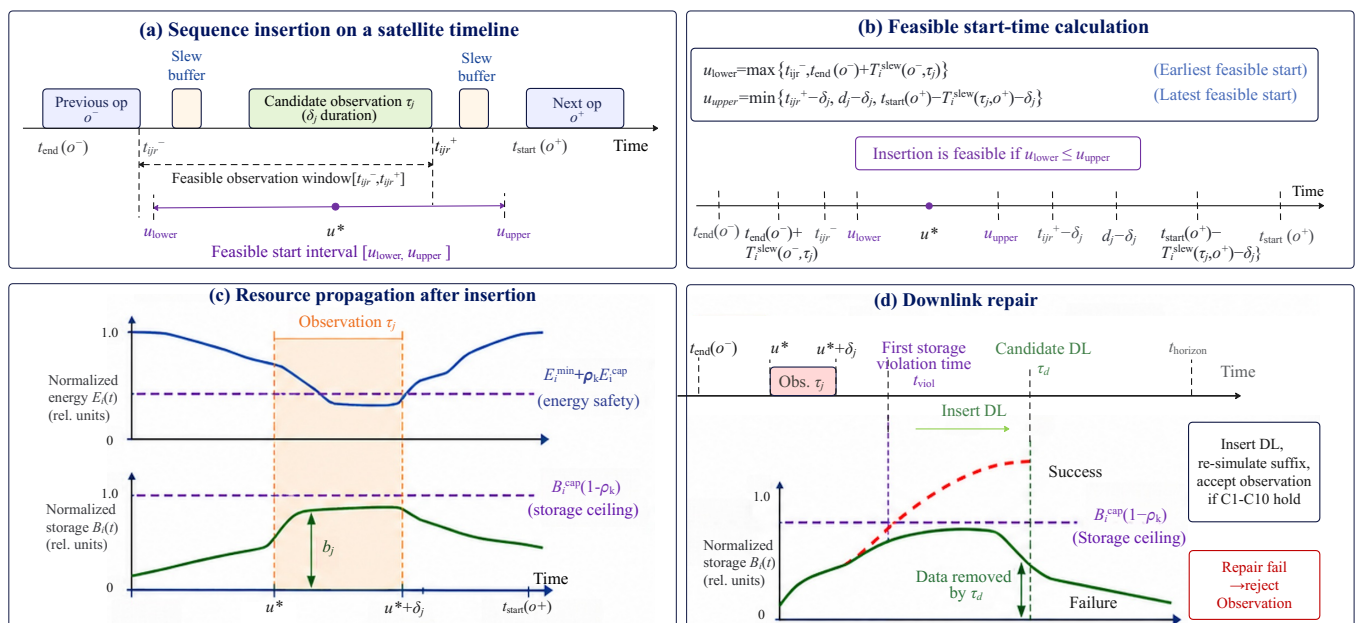


Figure 3. Observation insertion and bounded repair. This figure summarizes the operation-sequence mechanisms that preserve feasibility: the scheduler computes a feasible start interval from neighboring operations, propagates energy and storage along the affected suffix, and uses downlink repair only when a storage violation can be reduced by an available TTW.

5. Training and Online Inference

The scheduler is treated as part of the environment transition. During training, each orbital-group actor samples a raw Gaussian action, the bounded transformation in (45) and (46) converts it into scheduler preferences, and the deterministic scheduler returns $\Pi_t = \mathfrak{S}(s_t, a_t)$.

The factorized policy is

$$\pi_{\psi,\theta}(\tilde{a}_t|s_t) = \prod_{k=1}^K \mathcal{N}(\tilde{a}_k; \mu_\theta(o_k^t), \text{diag}(\sigma_\theta^2(o_k^t))), \quad (57)$$

where o_k^t is the graph readout in (44). The actor outputs (μ_k^t, ζ_k^t) and uses

$$\sigma_k^t = \exp(\text{clip}(\zeta_k^t, \log \sigma_{\min}, \log \sigma_{\max})) \quad (58)$$

during training. The clipping stabilizes exploration of the raw preference action; online inference uses the mean action and therefore does not sample from this Gaussian. A centralized critic estimates the team value from a global graph readout.

Given the scheduler mapping in (23), MAPPO optimizes

$$\max_{\psi,\theta} \mathbb{E} \left[\sum_{t=0}^{T-1} \gamma^t R_t \mid a_t \sim \pi_{\psi,\theta}, \Pi_t = \mathfrak{S}(s_t, a_t) \right]. \quad (59)$$

The shared reward uses the same two-weight structure as the rolling objective:

$$R_t = R_t^{\text{val}} - \lambda_{\text{op}} R_t^{\text{op}}, \quad (60)$$

$$R_t^{\text{val}} = \sum_{j \in \mathcal{C}_t^{\text{new}}} \bar{p}_j \left(1 + \lambda_u \kappa_j \exp \left[-(t_j^{\text{first}} - a_j) / \tau_u \right] \right), \quad (61)$$

$$R_t^{\text{op}} = \Psi_{\text{res}}(\Pi_t) + (1 - J_t) + \Psi_{\text{rev}}(\Pi_t). \quad (62)$$

The value part of the reward is triggered by newly observation-completed tasks, while the operational part carries energy, storage, age-weighted untransmitted-data, load-balance, and revision effects. Thus the strengthened observation-data model changes only how the returned sequence is scored and evaluated; it does not require gradients through the scheduler, additional actor heads, or a modified action map.

The policy and critic are updated with the standard clipped MAPPO objective using generalized advantage estimation,

$$\min_{\psi,\theta,\phi} -L_{\text{MAPPO}}^{\text{clip}}(\psi, \theta) + c_v L_V(\phi) - c_e \mathcal{H}(\pi_{\psi,\theta}). \quad (63)$$

No gradient is propagated through the deterministic scheduler. At online inference, the actor uses the mean raw action, transforms it into bounded preferences, and calls the same scheduler used during training.

Table 2 lists the implementation parameters fixed before evaluation. The values were selected to keep the actor compact relative to the deterministic scheduler: the learned output remains eight-dimensional per orbital group, while graph density and repair iterations are bounded explicitly.

Table 2. Fixed GRIP-MAPPO implementation parameters.

Item	Fixed Setting
Graph encoder	Hidden width $d = 128$; $L_G = 3$ relation-aware graph layers; 4 attention heads; FFN width 256; dropout 0.10.
Graph truncation	Retained-neighbor budgets: satellite-neighbor $q = 4$, satellite-task $p = 36$, task-conflict $c_\tau = 16$, and satellite-ground $r_g = 10$.
Orbital groups	Plane-based actors; the reference case has $K = 12$ groups with six satellites per group.
Preference action	Eight raw coordinates per group; $\sigma_{\min} = 0.05$, $\sigma_{\max} = 0.60$, $\beta_{\min} = 0.05$, $\beta_{\max} = 0.45$, and $\rho_{\max} = 0.15$.

Table 2. Cont.

Item	Fixed Setting
MAPPO loss	$\gamma = 0.995$, $\lambda_{\text{GAE}} = 0.95$, PPO clip 0.20, value coefficient $c_v = 0.5$, entropy coefficient decayed from 10^{-2} to 2×10^{-3} .
Optimizer	Adam with learning rate 3×10^{-4} and $\epsilon = 10^{-5}$; 4 PPO epochs; 8 minibatches; maximum gradient norm 0.5.
Rollouts and budget	48 rolling epochs per mission-day rollout; 8 parallel rollouts per PPO update, i.e., 384 epoch transitions before minibatch splitting; 5000 mission-day episodes per seed; five seeds; validation every 50 episodes.
Scheduler limits	$U_k \leq 3$ start trials per gap; $R_{\text{max}} = 3$ transmission-repair iterations; 60 s resource-check spacing; final C1–C10 audit.
Reward weights	$\lambda_\mu = 1.0$, $\tau_u = 2$ h, $\lambda_{\text{op}} = 0.12$, $\alpha_E = 1.0$, $\alpha_B = 1.0$, and $\alpha_D = 0.6$.

6. Computational Complexity

The online cost is counted after observation time windows (OTWs) and TTWs have been generated for the current epoch. Let $|V| = N + M + L$, $|E|$ be the retained edge count, L_G the number of graph layers, d the hidden width, K the number of orbital groups, C_k the retained candidate count for group k , I_k the average number of insertion gaps, U_k the number of tested start times per gap, and L_k the affected suffix length.

The graph encoder has complexity

$$T_{\text{enc}} = O(L_G(|E|d + |V|d^2)), \quad (64)$$

where the retained edge count is controlled by the truncation budgets:

$$|E| = O(Nq + Np + Mc_\tau + Nr_g). \quad (65)$$

Here q , p , c_τ , and r_g denote the satellite-neighbor, satellite-task, task-task, and satellite-ground retention budgets. For group k , observation insertion costs

$$T_k^{\text{obs}} = O(C_k \log C_k + C_k I_k U_k L_k), \quad (66)$$

because candidates are sorted and each accepted trial triggers suffix propagation. Bounded downlink repair contributes

$$T_k^{\text{dl}} = O(F_k R_{\text{max}} D_k I_k^{\text{dl}} V_k L_k^{\text{dl}}), \quad (67)$$

where F_k is the number of storage-triggered repair events, R_{max} is the repair iteration limit, D_k is the retained TTW count, and V_k is the number of tested downlink durations or start offsets. Duplicate resolution costs

$$T_{\text{dup}} = O\left(\sum_j a_j \log a_j + R_{\text{dup}} \bar{L}\right), \quad (68)$$

where a_j is the number of accepted observations for task τ_j before duplicate removal. With group-parallel construction, the wall-clock online cost is approximately

$$T_{\text{online}} = T_{\text{enc}} + O\left(\max_k (T_k^{\text{obs}} + T_k^{\text{dl}}) + T_{\text{dup}} + KC_{\text{actor}}\right). \quad (69)$$

The memory cost is

$$M_{\text{online}} = O(|V|d + |E|d_e + \sum_i |\pi_i^t|). \quad (70)$$

The transmission-status evaluation adds only a linear pass over generated observation-data items and accepted downlink operations,

$$T_{\text{trans}} = O\left(\sum_i |\pi_i^t| + \sum_i |\mathcal{Z}_i(t + \Delta H)|\right), \quad (71)$$

which is dominated by suffix propagation in the tested settings.

In the evaluated implementation, $U_k \leq 3$ and $R_{\text{max}} = 3$. Runtime is therefore dominated by retained graph edges and suffix propagation, not by the eight-dimensional actor output.

7. Experiments

We evaluate GRIP-MAPPO in a public-data-anchored rolling-horizon LEO constellation benchmark designed to isolate observation contention, downlink scarcity, onboard resource pressure, communication uncertainty, and emergency task bursts. The benchmark is controlled in the statistical sense: all methods face paired instances with the same OTWs, TTWs, task arrivals, initial resources, locked-operation histories, and disturbance realizations. Its orbital and ground-network geometry is not an arbitrary random scene; OTWs and TTWs are generated from SGP4-compatible propagated orbital elements and ESTRACK-derived station coordinates documented in the accompanying experiment protocol file.

7.1. Experimental Setup

The satellite population follows a reproducible Walker-like LEO construction anchored to public Earth-observation GP/TLE data. Specifically, the altitude–inclination envelope follows the public CelesTrak Planet group and the CelesTrak GP data interface [51,52]. SGP4-compatible mean-element propagation is used for OTW and TTW generation [53]. The ground network contains 12 globally distributed stations derived from the ESA/NAIF ESTRACK topocentric frame kernel [54]. The complete station coordinates and link-generation settings are fixed in the Supplementary Materials. Table 3 summarizes the core simulation settings.

For the reference scale, the constellation has 12 orbital planes with six satellites per plane. For all scales, RAANs are uniformly spaced, satellites are uniformly spaced in mean anomaly within each plane, and the Walker phasing factor is fixed to $F = 1$. OTW and TTW boundaries are detected on a 30 s ephemeris grid and refined to 1 s resolution. Each station has one reception channel at $N = 24$ and $N = 72$, two channels at $N = 144$, and four channels at $N = 288$. Within one channel, one satellite at most may downlink at a time.

Each satellite has normalized energy capacity 1.0, safety floor $E_i^{\text{min}} = 0.10$, and nominal storage capacity $B_i^{\text{cap}} = 512$ Gbit unless modified by the scenario. Targets are visible when the required off-nadir angle is at most 35° . Observation durations are sampled from 12–45 s. Normal tasks have value $p_j \sim U[1, 3]$, deadline slack $U[4, 12]$ h, coverage multiplicity one with probability 0.9 and two otherwise, and data volume $U[0.8, 4.0]$ Gbit. Urgent tasks have value $U[6, 10]$, deadline slack $U[1, 3]$ h, multiplicity one, and data volume $U[1.0, 5.2]$ Gbit.

Table 3. Core simulation settings.

Item	Setting
Simulation epoch	2026-01-01 00:00:00 UTC
Orbit envelope	Public-data-anchored circular sun-synchronous Walker-like LEO; altitude 500 km, inclination 97.4° , eccentricity 0.001
Constellation scales	$N \in \{24, 72, 144, 288\}$; the 72-satellite case is the reference scale
Ground network	12 fixed stations; minimum elevation 10° ; contacts shorter than 60 s discarded
Rolling horizon	24 h episode, 48 epochs, 4 h look-ahead, 30 min committed segment
Task scale	Reference nominal count $M_0 = 1200$; other scales use $M_0(N) = \lfloor 16.7N \rfloor$
Data split	Per scenario: 600 training seeds, 60 validation seeds, and 60 held-out test seeds

7.2. Operating Regimes

Table 4 summarizes the six fixed operating regimes used for training-curve evaluation and robustness testing. They are controlled instance generators rather than post hoc labels. S1 is the balanced reference condition. S2–S5 isolate one dominant pressure source each, while S6 combines emergency arrivals with downlink and communication stress. The detailed task-clustering, burst, resource, and link perturbation parameters are fixed in the Supplementary Materials.

Table 4. Operating regimes used in the main experiments. Multipliers are relative to the nominal reference setting.

ID	Main Stress	Obs. Load	DL Load	Resource State	Additional Controls
S1	Balanced nominal	$1.0M_0$	data $\times 1.0$, rate $\times 1.0$	$B_0 = 20\text{--}40\%$, $E_0 = 65\text{--}95\%$	10% clustered tasks, 5% urgent tasks, no link noise or outage
S2	Observation contention	$1.6M_0$	data $\times 1.0$, rate $\times 1.0$	same as S1	60% clustered tasks from 2–3 hotspots; shorter normal deadlines
S3	Downlink bottleneck	$1.1M_0$	data $\times 1.6$, rate $\times 0.70$	$B_0 = 35\text{--}55\%$, $E_0 = 60\text{--}90\%$	no link noise or outage; moderate observation pressure
S4	Resource-tight satellites	$1.2M_0$	data $\times 1.0$, rate $\times 1.0$	storage capacity $\times 0.80$, $B_0 = 45\text{--}65\%$, $E_0 = 45\text{--}65\%$	charging $\times 0.80$; slew time and slew energy $\times 1.30$
S5	Communication uncertainty	$1.1M_0$	data $\times 1.1$, stochastic rate	$B_0 = 30\text{--}50\%$, $E_0 = 60\text{--}90\%$	log-normal link factor with CV 0.30; one 90 min station outage
S6	Emergency combined stress	$1.3M_0$ plus urgent burst	data $\times 1.3$, rate $\times 0.80$	$B_0 = 35\text{--}55\%$, $E_0 = 55\text{--}80\%$	urgent burst of $0.15M_0$ tasks; link CV 0.25; one 60 min outage

7.3. Compared Methods and Metrics

The comparison separates methods according to their experimental role rather than treating all baselines as directly comparable online alternatives. Table 5 lists three roles. Direct online scheduling alternatives include EDF, profit-density greedy, tuned constructive heuristic, vector-state MAPPO, homogeneous-graph MAPPO, and GRIP-MAPPO; these methods return one rolling-epoch schedule under the same online replanning setting. Long-budget LNS is included only as a search-quality reference because it performs iterative destroy-and-repair search under a substantially larger, scale-dependent wall-clock budget. CP-SAT/MILP is used only for small-scale calibration where exact or bounded search remains tractable. Therefore, the reported experiments should not be read as one equal-runtime online leaderboard: direct online comparisons are made among the online schedulers, whereas LNS and CP-SAT/MILP provide reference points for search quality and small-scale calibration, respectively. Non-learning baselines do not appear in learning-curve figures because they do not have training episodes.

Table 5. Compared method groups and their experimental roles. Direct online alternatives are evaluated under the online replanning setting; long-budget LNS is a search-quality reference; CP-SAT/MILP is a small-scale calibration reference.

Group	Methods and Role
Heuristics	EDF [1,2], profit-density greedy [9,10], and tuned heuristic methods [19,20] provide direct online scheduling references under the same rolling-horizon evaluation setting.
Long-budget search reference	Long-budget LNS [21] uses the same insertion, repair, and validation primitives as the proposed scheduler, but performs iterative destroy-and-repair search under a scale-dependent wall-clock budget. It is reported as a search-quality reference, not as an equal-runtime online baseline.
Learning-based online alternatives	Vector-state MAPPO [32,40] removes graph messages; homogeneous-graph MAPPO [40–42] removes relation typing; GRIP-MAPPO is the full proposed method. These methods are compared as online learning schedulers.
Small-scale exact/bounded calibration	CP-SAT/MILP [13–15] is run only on 6-, 12-, and 24-satellite instances to report optimality or best-bound references where exact or bounded optimization remains tractable. It is not used as a large-scale online baseline.

Training behavior is evaluated by scenario-wise normalized episodic return:

$$\tilde{R}_{s,e} = \frac{R_{s,e} - R_s^{\text{low}}}{R_s^{\text{high}} - R_s^{\text{low}} + 10^{-8}}, \quad (72)$$

where R_s^{low} is the mean EDF return on the validation seeds and R_s^{high} is the mean long-budget LNS return on the same validation seeds. The same normalization pair is used for all algorithms and seeds within scenario s , and values are not clipped. Final schedules are evaluated by weighted observation completion (WOC), transmitted-data fraction (TDF), urgent-task completion (UTC), online replanning time per rolling epoch, and the independent C1–C10 feasibility audit. Diagnostics such as peak storage ratio, minimum energy margin, repair attempts, repair success rate, and infeasible-candidate rejection rate are recorded for ablation and supplementary reporting.

7.4. Training, Fairness, and Statistics

All MAPPO variants use the reward in (60)–(62), the same eight-dimensional scheduler action map, and the implementation parameters in Table 2. A mixed-scenario training distribution samples S1–S6 uniformly, producing one deployable policy per learning method rather than six scenario-specialized policies. Each learning method is trained for 5000 mission-day episodes per seed using five independent seeds. A PPO update consumes eight complete mission-day rollouts, each rollout contains 48 rolling-epoch transitions, and check points are evaluated every 50 training episodes.

All paired comparisons use identical test seeds, task arrivals, OTWs, TTWs, initial resources, link multipliers, outage intervals, and urgent-burst realizations. Hyperparameters and tuned heuristic weights are selected only on validation seeds. Main tables report mean $\pm$ 95% confidence interval over paired held-out test instances, together with Holm–Bonferroni-corrected significance markers for the GRIP-MAPPO quality metrics. GRIP-MAPPO is compared against the strongest non-GRIP online baseline in each experiment using a two-sided paired Wilcoxon signed-rank test with Holm–Bonferroni correction; the long-budget LNS results are retained as search-quality references rather than as online-deployment comparators, and no significance claim is made against LNS unless explicitly stated. All experiments were implemented in Python (v3.10) with PyTorch (v2.10.0). Runtime is measured on an Intel Xeon Gold 6338 workstation with 256 GB RAM and one NVIDIA RTX 4090 GPU; CPU-side scheduling, feasibility checking, and LNS run with one process and one thread per instance.

7.5. Learning Dynamics Across Six Operating Regimes

Figure 4 reports the training dynamics of GRIP-MAPPO and two learning ablations under the six operating regimes. Each curve shows the normalized episodic return averaged over five independent training seeds, and the shaded region denotes the standard error. A 50-episode moving average is used only for visualization.

Several observations can be made from Figure 4. First, all three learning variants improve from near-zero normalized return, indicating that the bounded preference-action interface provides a trainable control signal even though the final operation sequence is constructed by a deterministic feasibility-preserving scheduler. Second, GRIP-MAPPO achieves the fastest and most stable improvement in all six regimes. In the balanced nominal case, the proposed method reaches a high-return plateau earlier than both ablations, suggesting that the heterogeneous satellite–task–ground representation accelerates policy learning when resource conflicts are moderate. In the observation-contention regime, homogeneous-graph MAPPO remains competitive during part of training, but GRIP-MAPPO still obtains the highest final return, showing that relation typing is useful even when the dominant difficulty is dense target competition rather than downlink scarcity.

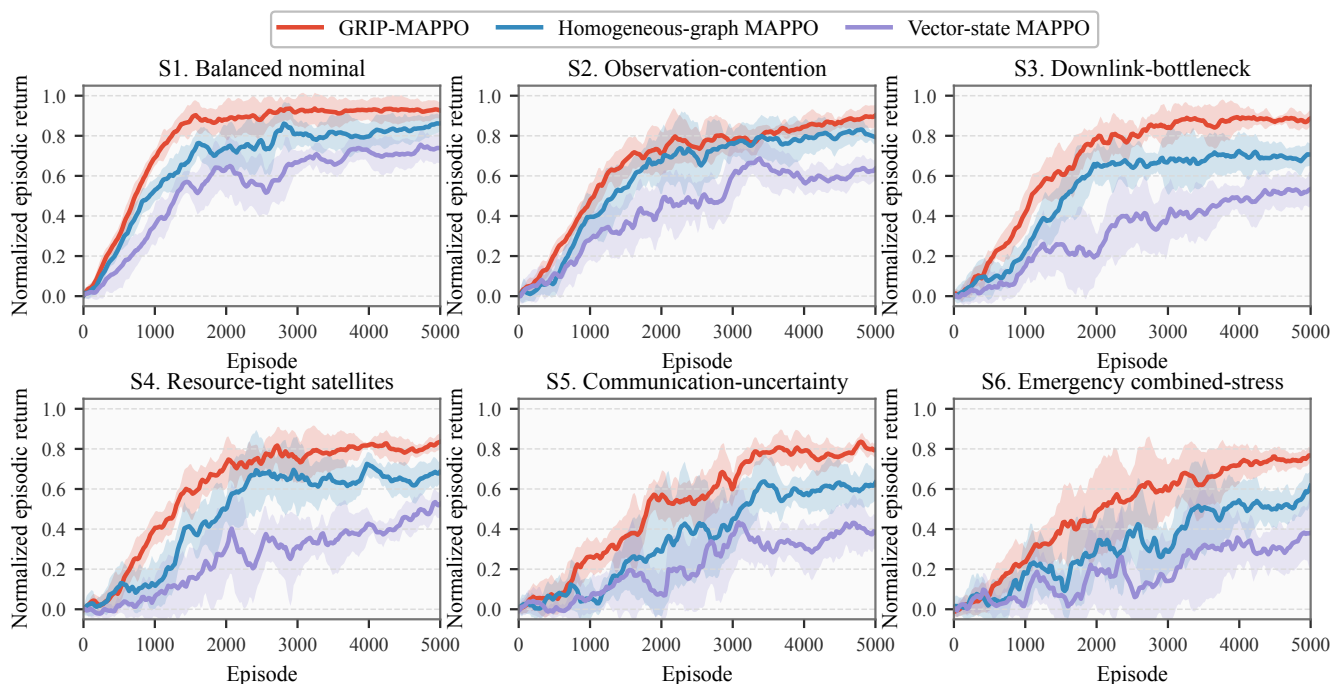


Figure 4. Learning dynamics of normalized episodic return across six operating regimes. Solid curves show the mean over five independent training seeds, and shaded bands show the standard error. GRIP-MAPPO consistently reaches the highest return across all regimes. Its advantage is moderate in the balanced and observation-contention regimes, but becomes more pronounced under downlink bottlenecks, resource-tight satellites, communication uncertainty, and emergency combined stress.

The gap becomes larger in the more constrained regimes. Under the downlink-bottleneck and resource-tight settings, GRIP-MAPPO maintains a clear advantage over homogeneous-graph MAPPO and vector-state MAPPO, which indicates that explicitly encoding satellite–ground contacts, onboard storage pressure, and resource-dependent relations helps the policy assign more appropriate scheduling preferences before deterministic repair and validation are invoked. Under communication uncertainty and emergency combined stress, all methods exhibit larger variance because link perturbations, station outages, urgent bursts, and tight deadlines make the rolling-horizon return more sensitive to early scheduling decisions. Nevertheless, GRIP-MAPPO remains consistently above the

two ablations after convergence. This trend suggests that the proposed graph-relational insertion-preference policy does not merely fit the nominal scenario, but learns a more transferable priority surface across heterogeneous operating pressures.

The two ablations clarify the source of the improvement. Vector-state MAPPO removes relational message passing and learns more slowly, especially in S3–S6, where local satellite statistics alone are insufficient to represent shared downlink contention and burst-induced task coupling. Homogeneous-graph MAPPO improves over the vector-state baseline but remains below GRIP-MAPPO in most regimes, showing that using graph structure without relation-specific modeling only partially captures the scheduling state. These results support the design choice of combining typed graph encoding, orbital-group actors, and a deterministic constrained scheduler: the neural policy learns adaptive preferences, while feasibility remains enforced by explicit insertion, propagation, repair, and audit procedures.

7.6. Reference-Scale Scheduling Performance

Table 6 reports the held-out reference-scale results for $N = 72$ in the balanced nominal regime. All entries use the paired test instances and the same C1–C10 audit, but they do not serve the same evaluation role. EDF, profit-density greedy, tuned heuristic, vector-state MAPPO, homogeneous-graph MAPPO, and GRIP-MAPPO are interpreted as online schedulers under rolling replanning. Long-budget LNS is included as a search-quality reference with a substantially larger wall-clock budget, rather than as an equal-runtime online alternative. Lightweight greedy methods are the fastest, but they leave substantial observation value and transmitted data unexploited. GRIP-MAPPO requires only a small constant-factor increase in online time over EDF because online inference consists of graph construction, one bounded actor pass, and deterministic insertion-and-validation. At the same time, it improves WOC by 23.3 percentage points over EDF and by 5.4 points over homogeneous-graph MAPPO. The long-budget LNS reference reaches competitive quality, but its per-epoch runtime is two orders of magnitude larger; it therefore indicates the quality attainable by extended search rather than the runtime of an online deployment baseline. Accordingly, the small absolute differences between GRIP-MAPPO and long-budget LNS in this table should be interpreted as near-parity at very different runtime budgets, not as evidence that GRIP-MAPPO is statistically superior to the long-budget search reference.

Table 6. Reference-scale scheduling performance at $N = 72$ under S1. WOC, TDF, and UTC are percentages; runtime is wall-clock online replanning time per rolling epoch.

Method	WOC $\uparrow$	TDF $\uparrow$	UTC $\uparrow$	Time $\downarrow$	Audit
EDF greedy	64.2 $\pm$ 1.8	61.2 $\pm$ 2.2	56.6 $\pm$ 2.6	0.11 s	0% viol.
Profit-density greedy	67.6 $\pm$ 1.7	59.8 $\pm$ 2.3	60.4 $\pm$ 2.5	0.14 s	0% viol.
Tuned constructive heuristic	74.8 $\pm$ 1.5	70.2 $\pm$ 1.9	68.4 $\pm$ 2.2	0.20 s	0% viol.
vector-state MAPPO	75.6 $\pm$ 1.6	69.1 $\pm$ 2.1	70.6 $\pm$ 2.4	0.16 s	0% viol.
Homogeneous-graph MAPPO	82.1 $\pm$ 1.3	74.7 $\pm$ 1.7	75.6 $\pm$ 1.9	0.23 s	0% viol.
Long-budget LNS	86.9 $\pm$ 1.2	80.6 $\pm$ 1.6	81.2 $\pm$ 1.8	48.0 s	0% viol.
GRIP-MAPPO	87.5 $\pm$ 1.1 ***	81.2 $\pm$ 1.5 ***	82.1 $\pm$ 1.7 ***	0.27 s	0% viol.

Note: Superscripts on GRIP-MAPPO quality metrics denote Holm–Bonferroni-corrected two-sided paired Wilcoxon signed-rank tests against the strongest non-GRIP online baseline under the same paired test instances: * $p_{\text{adj}} < 0.05$, ** $p_{\text{adj}} < 0.01$, and *** $p_{\text{adj}} < 0.001$. Long-budget LNS is included as a search-quality reference and is not used as the online-deployment comparator. The markers therefore should not be read as significance markers against long-budget LNS.

The TDF column is particularly important for this problem setting. Profit-density greedy improves WOC over EDF, but its TDF is slightly lower because it favors compact high-value observations without explicitly adapting to downlink pressure. GRIP-MAPPO instead uses satellite–ground relations, storage pressure, and the downlink-aware resource

term to preserve observation value while also increasing transmitted-data fraction. The zero-violation audit should be interpreted as a schedule-level certificate produced by the shared deterministic scheduler and final C1–C10 audit, rather than as a feasibility guarantee learned by the neural policy itself. Failed trial insertions are rejected before commitment, or rolled back if a later audit fails, and therefore are not counted as final audit violations of the returned schedule. These rejected trials may still affect runtime and solution quality through additional insertion, propagation, and repair checks, or by leaving some valuable candidates uncommitted. Thus, the learning contribution of GRIP-MAPPO is improved feasible-schedule quality under the same feasibility-preserving scheduler interface, not feasibility enforcement alone.

7.7. Robustness Under Operating Stress

Figure 5 compares WOC and TDF across the six operating regimes. GRIP-MAPPO consistently achieves the best performance among the direct online scheduling alternatives. The margin is moderate in S1 and S2, where the main difficulty is ordinary target selection or dense observation contention. The margin grows in S3–S6, where downlink scarcity, resource pressure, link uncertainty, and emergency bursts make local dispatching rules less reliable.

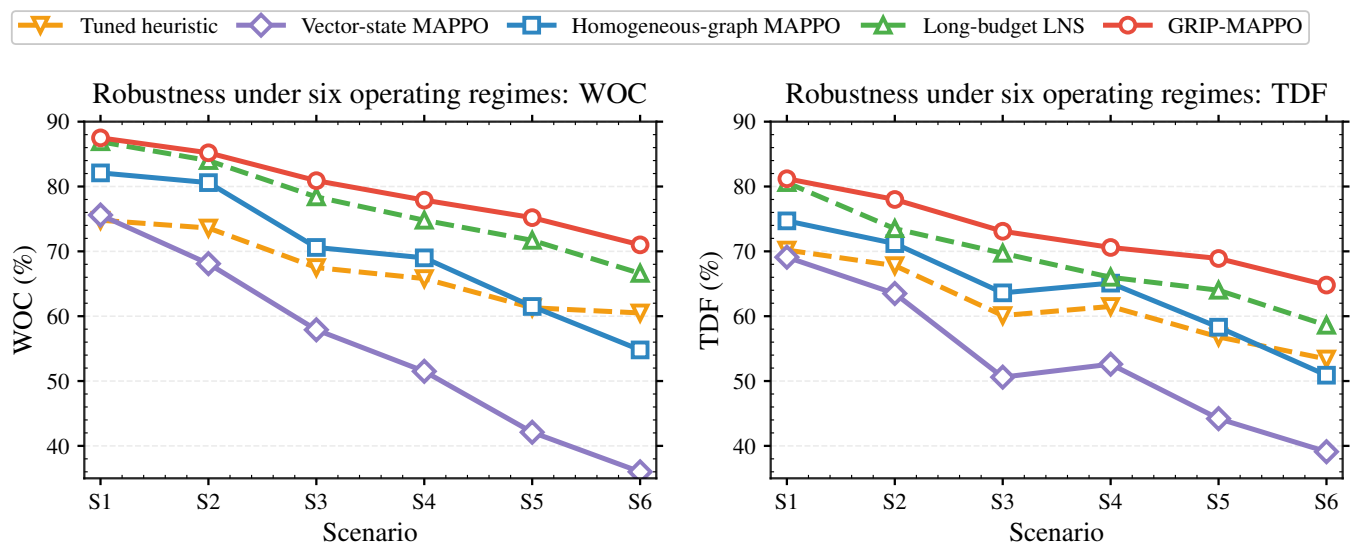


Figure 5. Robustness results at $N = 72$. GRIP-MAPPO maintains the highest WOC and TDF across all regimes. The advantage is largest in downlink-bottleneck, communication-uncertainty, and emergency combined-stress settings, where relation-specific satellite–task–ground information and bounded downlink repair are most useful.

In S3, GRIP-MAPPO reaches 80.9% WOC and 73.1% TDF, compared with 70.6% and 63.6% for homogeneous-graph MAPPO. In S6, the gap widens to 16.2 WOC points and 13.9 TDF points. These results match the learning-curve evidence in Figure 4: the value of relation-specific encoding becomes more visible when feasible observations, storage relief, and urgent observation-response decisions are coupled. Long-budget LNS remains competitive, but its search cost is not comparable to the learned online schedulers; GRIP-MAPPO provides higher S6 WOC while retaining sub-second per-epoch inference at the reference scale.

7.8. Scalability Analysis

Figure 6 evaluates scalability from 24 to 288 satellites. The quality plots use S1 and S6 to separate nominal scaling from stress scaling, and the runtime plot reports online

replanning time per rolling epoch. The results show three regimes. EDF and tuned constructive heuristics are fastest but lose substantial WOC as coupling increases. Long-budget LNS provides a strong search reference at small scale but becomes minutes-to-hours for full mission-day rollouts as N grows. GRIP-MAPPO stays within a small constant factor of greedy construction while giving the best or near-best WOC.

At $N = 288$, GRIP-MAPPO obtains 82.0% WOC in S1 and 64.6% WOC in S6. The corresponding measured per-epoch runtime is 2.33 s, compared with 0.81 s for EDF and 420 s for long-budget LNS under a 720 s search cap. Thus, the method does not rely on expensive iterative improvement to scale. Its runtime increases mainly because graph construction, retained OTW/TTW edges, and suffix propagation grow with the number of satellites and tasks, consistent with the complexity analysis in Section 6.

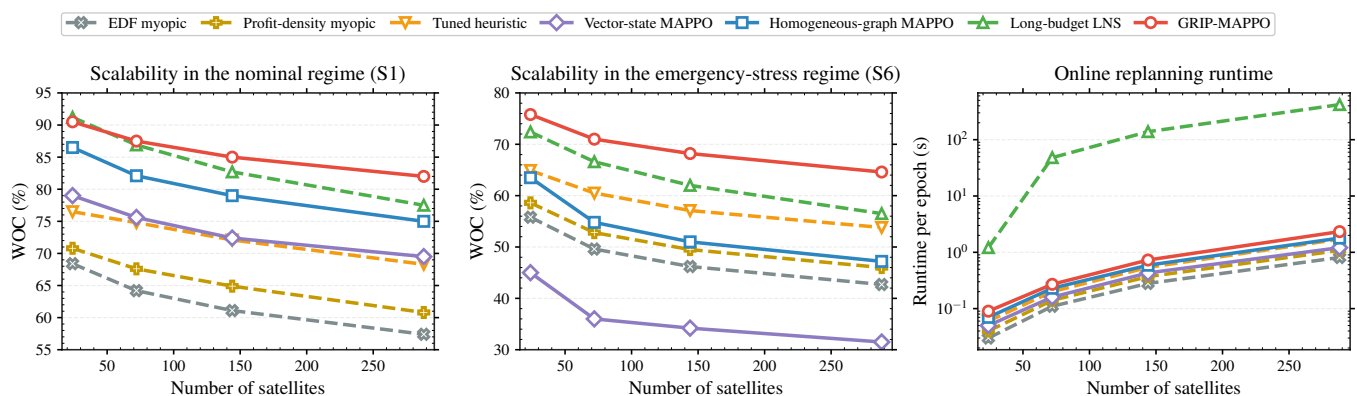


Figure 6. Scalability from 24 to 288 satellites. The two quality figures separate nominal scaling from emergency-stress scaling, and the runtime figure reports online replanning time per rolling epoch on a log scale. GRIP-MAPPO is not the fastest method, but it lies on the best quality–runtime frontier: it is roughly 2–3 $\times$ slower than EDF and orders of magnitude faster than long-budget LNS at medium and large scales.

7.9. Ablation Study

Table 7 links the empirical results to the main interface components highlighted above and isolates the contribution of each design choice. Removing relation typing reduces GRIP-MAPPO to the homogeneous-graph variant and causes a 5.4-point WOC loss in S1, a 10.3-point loss in S3, and a 16.2-point loss in S6. This confirms that the typed graph does more than increase model capacity: it separates OTW alternatives, task conflicts, satellite coordination, and TTW/downlink opportunities before the scheduler ranks candidates.

Table 7. Component ablation at $N = 72$. The full-model rows match the corresponding S1, S3, and S6 robustness results.

Variant	S1 WOC	S1 TDF	S3 WOC	S3 TDF	S6 WOC	S6 TDF
Full GRIP-MAPPO	87.5 **	81.2 **	80.9 ***	73.1 ***	71.0 ***	64.8 ***
Without heterogeneous relation types	82.1	74.7	70.6	63.6	54.8	50.9
Without satellite–ground/downlink edges	84.6	74.2	72.8	59.8	61.7	51.2
Without orbital-group actors	83.8	77.4	74.8	67.8	63.2	57.0
Without bounded downlink repair	82.3	65.1	62.0	42.5	50.1	38.0

Note: Superscripts on the full-model quality metrics denote Holm–Bonferroni-corrected paired Wilcoxon tests against the strongest ablated variant for the same scenario and metric: * $p_{\text{adj}} < 0.05$, ** $p_{\text{adj}} < 0.01$, and *** $p_{\text{adj}} < 0.001$.

The downlink-specific ablations explain the stress-regime gains. Removing satellite–ground/downlink edges has a modest WOC effect in S1 but sharply lowers TDF in S3 and

S6, showing that downlink opportunities must be visible to the actor before the deterministic scheduler is called. Disabling bounded downlink repair is the most damaging change under transmission stress: S3 TDF drops from 73.1% to 42.5%, and S6 TDF drops from 64.8% to 38.0%. Replacing orbital-group actors also degrades performance, especially in S6, because a single global policy readout weakens localized coordination among satellites that share targets and ground contacts.

To illustrate the interpretability of the eight-dimensional preference interface, Table 8 adds a post-training coordinate sensitivity analysis. This coordinate-level ablation is conducted in a post-training manner: the trained GRIP-MAPPO policy is kept fixed, one preference coordinate is replaced by its neutral mixed-scenario mean at inference time, and the first six ranking weights are re-normalized when one of them is perturbed. The purpose is to show which preference channels most strongly affect the returned feasible schedules under the same scheduler and final audit.

Table 8. Post-training coordinate-level ablation of the eight-dimensional preference vector at $N = 72$. Each row neutralizes one coordinate at inference time while keeping the trained policy, graph encoder, scheduler, and feasibility audit unchanged.

Preference Perturbation	S1 WOC	S1 TDF	S3 WOC	S3 TDF	S6 WOC	S6 TDF
Full GRIP-MAPPO	87.5	81.2	80.9	73.1	71.0	64.8
Neutralized profit weight ω_p	85.7	79.6	78.6	70.7	68.6	62.5
Neutralized urgency weight ω_d	86.4	80.1	78.9	70.9	64.9	58.8
Neutralized imaging-quality weight ω_q	86.1	80.0	79.1	71.2	68.9	63.0
Neutralized energy-cost weight ω_e	85.9	79.1	77.2	69.0	65.6	59.7
Neutralized memory/downlink weight ω_b	85.5	76.8	75.1	65.6	63.6	55.1
Neutralized load-balance weight ω_l	84.7	78.4	77.6	69.7	65.8	59.4
Fixed acceptance threshold β	85.2	79.0	77.7	69.4	66.6	60.5
Fixed reservation margin ρ	86.0	77.9	76.2	67.6	64.4	57.8

The sensitivity pattern is consistent with the component ablations. Memory/downlink pressure and the reservation margin have the largest effect on TDF in S3 and S6, matching the importance of satellite-ground edges and bounded downlink repair. Urgency has the largest WOC effect in S6, where emergency tasks and tight deadlines dominate. Load balance and the acceptance threshold mainly affect WOC by changing how aggressively candidates are admitted across locally congested satellites. Profit and imaging quality remain useful for value ordering, but their neutralization is less damaging than removing downlink-aware or stress-adaptive preference channels because the deterministic scheduler still validates candidate feasibility after ranking.

7.10. Transmission-Constrained Sensitivity

Table 9 reports a controlled transmission-stress sweep at $N = 72$. The sweep jointly increases generated data volume and decreases link rate so that the third row matches the S3 downlink-bottleneck regime. This table connects the S1 nominal result, the S3 stress result, and the downlink-repair ablation in one consistency check.

As transmission stress increases, all methods lose WOC and TDF, but the degradation is slowest for GRIP-MAPPO. The gap between GRIP-MAPPO and homogeneous-graph MAPPO grows from 5.4 WOC points in the nominal row to 12.9 points in the severe row. The no-repair variant deteriorates much faster, confirming that the deterministic repair loop is essential when storage violations appear before the next useful TTW.

Table 9. Transmission-stress sensitivity at $N = 72$. The S3-equivalent row uses the full S3 downlink-bottleneck generator, including data $\times 1.6$, link rate $\times 0.70$, task-load, and initial-resource settings.

Stress Level	Data	Rate	GRIP WOC	GRIP TDF	Hom WOC	Hom TDF	LNS WOC	LNS TDF	No-Repair WOC	No-Repair TDF
Nominal	$\times 1.0$	$\times 1.00$	87.5 ***	81.2 ***	82.1	74.7	86.9	80.6	82.3	65.1
Mild	$\times 1.3$	$\times 0.85$	84.2 ***	77.2 ***	75.9	68.2	82.0	74.8	71.1	52.0
S3-equivalent	$\times 1.6$	$\times 0.70$	80.9 ***	73.1 ***	70.6	63.6	78.4	69.7	62.0	42.5
Severe	$\times 2.0$	$\times 0.55$	75.2 ***	66.8 ***	62.3	54.5	71.3	61.0	50.3	33.7

Note: Superscripts on GRIP quality metrics denote Holm–Bonferroni-corrected paired Wilcoxon tests against the strongest non-GRIP online comparator in the same stress row: * $p_{\text{adj}} < 0.05$, ** $p_{\text{adj}} < 0.01$, and *** $p_{\text{adj}} < 0.001$. LNS columns are reported as long-budget search references.

7.11. Small-Scale Reference Calibration

Table 10 reports small-scale calibration against CP-SAT/MILP. The comparison is expressed as relative objective value with the CP-SAT/MILP reference incumbent normalized to 100. The 6-satellite cases are solved to optimality; the 12-satellite cases are solved to a negligible solver-reported bound gap; and the 24-satellite cases use the best incumbent after the 1800 s limit. Therefore, the last row reports relative loss to the best available incumbent rather than a certified optimality gap. This calibration is not intended to replace the large-scale rolling-horizon evaluation; it checks whether the proposed online method remains close to the strongest tractable references under the same modeled constraints.

Table 10. Small-scale calibration against CP-SAT/MILP references. Objective values are normalized by the CP-SAT/MILP reference incumbent for each paired instance; the 24-satellite row is not a certified optimality-gap claim. Runtime is the complete 48-epoch rollout time of GRIP-MAPPO.

Scale	CP Status	EDF	Tuned	LNS	GRIP	Rel. Loss	GRIP Time
6 sats	Proven opt.	79.2	92.8	98.2	96.1	3.9%	0.8 s
12 sats	$\leq 0.5\%$ gap	78.7	92.5	97.1	95.0	5.0%	1.7 s
24 sats	1800 s inc.	76.8	91.8	96.2	94.6	5.4%	4.1 s

GRIP-MAPPO remains within 3.9–5.4% of the CP-SAT/MILP reference incumbent while requiring at most 4.1 s for the complete 48-epoch small-scale rollout. Long-budget LNS is closer to the CP reference in this small regime, as expected from a search method with substantially more wall-clock budget. The calibration therefore supports the intended interpretation: GRIP-MAPPO is a fast feasibility-preserving online scheduler that remains close to the strongest tractable CP-SAT/MILP reference incumbents under the tested reduced-scale settings, while requiring orders-of-magnitude less computation time. This comparison should be interpreted as a practical quality–runtime calibration, especially for the 24-satellite cases where the reference is the best incumbent after the 1800 s limit.

7.12. Core-Hyperparameter Sensitivity

To confirm the robustness of the default implementation parameters, we conduct a concise sensitivity analysis under the most stressful S6 scenario. Two representative hyperparameters are tested: the resource-reservation threshold ρ_{max} and the graph-density multiplier η_g . The former controls the maximum reservation margin used by the deterministic scheduler, while the latter scales the retained-edge budgets (q, p, c_τ, r_g) in heterogeneous graph construction.

As shown in Figure 7, a smaller ρ_{max} slightly increases short-term WOC but reduces TDF. This indicates that aggressive observation acceptance may accumulate excessive onboard data and increase future downlink pressure. In contrast, an overly large ρ_{max} makes the scheduler more conservative, which reduces observation completion. The default

value $\rho_{\max} = 0.15$ therefore provides a balanced trade-off between observation completion and future resource protection.

For the graph-structure limit, overly sparse graphs remove useful observation-window, task-conflict, and transmission-window relations before scheduling, leading to lower WOC and TDF. Increasing graph density improves scheduling quality at first, but the improvement becomes marginal after the default setting $\eta_g = 1.00$. Meanwhile, runtime continues to increase as more graph edges are retained. Therefore, the default graph budgets $(q, p, c_\tau, r_g) = (4, 36, 16, 10)$ provide a practical balance between scheduling quality and online operation speed.

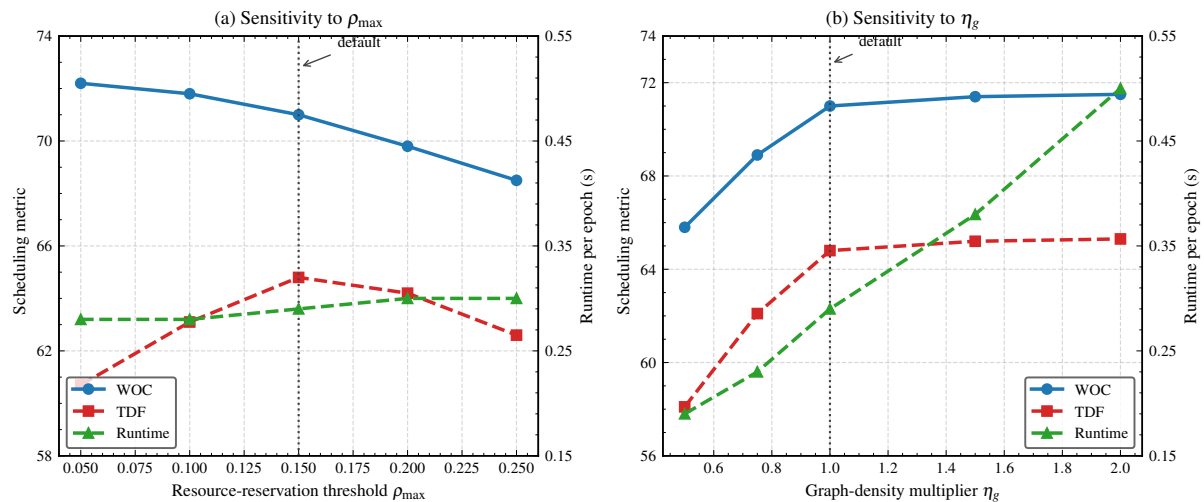


Figure 7. Core-hyperparameter sensitivity under S6. The tested parameters are the resource-reservation threshold $\rho_{\max}$ and the graph-density multiplier η_g . WOC and TDF evaluate scheduling quality, while runtime measures online operation speed.

8. Discussion

The results support three claims about the proposed interface. First, GRIP-MAPPO improves schedule quality without transferring constraint satisfaction to a neural network. In every reported setting, returned schedules pass the same C1–C10 audit. The actor shapes the candidate-priority surface, acceptance threshold, and reservation margin; timing, attitude transitions, resource propagation, downlink feasibility, duplicate resolution, and rolling commitment remain validated by the deterministic scheduler.

Second, the method is designed for an online quality–runtime trade-off. Greedy baselines are fastest but lose substantial WOC, TDF, and UTC. Long-budget LNS is a strong search-quality reference because it uses the same insertion, repair, and validation primitives with a much larger wall-clock budget; it should not be interpreted as an equal-runtime online baseline. GRIP-MAPPO adds bounded graph encoding and actor inference to the constructive scheduler and therefore stays within a small constant factor of greedy construction while approaching the long-budget reference at the reference scale and exceeding it only in those stressed or larger-scale settings where the reported table values show a clear advantage.

Third, the downlink-aware components matter most when observation choices and storage relief are coupled. In nominal settings, several methods can identify high-value observations. Under downlink bottlenecks, communication uncertainty, and emergency bursts, satellite–ground edges expose near-term storage relief to the actor, relation-specific messages prevent OTW, conflict, coordination, and TTW information from being collapsed into one generic edge type, and bounded downlink repair converts storage pressure into validated data-removal operations when a useful TTW exists.

The formulation targets acquisition-driven Earth-observation scheduling, where retained observations define the primary mission value and downlink supports storage feasibility, future observation opportunities, and residual-data-risk control. Delivery-driven missions can use the same observation–downlink sequence representation but should assign direct objective terms to downlink completion, delivered data value, or delivery latency. It tracks transmitted-data fraction and residual backlog to evaluate whether accepted observations can be supported by available TTWs, but the primary completion value remains observation-based. The experiments use a public-data-anchored deterministic benchmark, rather than a proprietary command-history replay or a flight-qualification campaign. Higher-fidelity orbit, link, attitude, power, or thermal models can replace the simplified components if they provide deterministic OTWs, TTWs, transition times, rates, and resource propagation to the scheduler. Finally, the method is not complete or globally optimal: graph truncation, greedy ranking, and bounded repair can exclude opportunities that a longer exact or neighborhood search may find. These limitations leave room for future scheduler variants with longer search budgets, operational flight-qualification studies, and higher-fidelity link and power models.

Practical Engineering Connection

In practical satellite mission operations, GRIP-MAPPO can be connected to the existing scheduling control system as a ground-side online scheduling and decision-support module, rather than as an autonomous flight-control component. The mission-planning system provides observation requests, task priorities, orbital ephemerides, observation time windows, and initial satellite resource states. The ground-station scheduling system provides transmission time windows, link-rate estimates, and channel-availability information. The onboard resource management unit reports telemetry on energy, storage, executed observations, and transmitted data. Based on these inputs, GRIP-MAPPO generates a feasible observation–downlink schedule through the deterministic scheduler, and only the locked execution segment is delivered to the command-generation system for uplink. After execution, updated satellite telemetry and ground-station transmission records are fed back to the next rolling scheduling epoch.

The practical deployment can be organized in four stages. First, the policy is trained and stress-tested offline using historical or simulated task streams. Second, the trained policy and deterministic scheduler are integrated with mission-planning, telemetry, and ground-station databases through standardized data interfaces. Third, shadow-mode validation is conducted by comparing GRIP-MAPPO schedules with operator-approved schedules without directly commanding satellites. Finally, the method can be used in supervised online scheduling, where the deterministic feasibility audit remains active and existing rule-based or optimization-based schedulers are retained as fallback mechanisms. This deployment mode allows the learned policy to improve scheduling quality while keeping hard feasibility verification in deterministic software.

9. Conclusions

This paper presented GRIP-MAPPO, a feasibility-preserving learning-guided framework for downlink-constrained rolling-horizon scheduling of LEO remote-sensing constellations. The formulation keeps observation tasks as the value-bearing mission requests while explicitly tracking onboard observation data, TTW-based transmission, age-weighted untransmitted backlog, energy and storage propagation, ground-station channel contention, and rolling command commitment. These additions do not enlarge the neural action interface: orbital-group actors output an eight-dimensional bounded preference

vector, and all transmission-status quantities are deterministic functions of the accepted operation sequence.

The method combines typed satellite–task–ground graph encoding, orbital-group MAPPO, and a deterministic insertion-and-validation scheduler. The learned policy adapts preferences over profit, urgency, imaging quality, energy cost, memory / downlink pressure, load balance, candidate acceptance, and reservation margin. The scheduler then performs observation insertion, suffix propagation, storage-triggered downlink repair, duplicate resolution, transmission-status evaluation, and final audit. The structural feasibility result shows that, under sound propagation and validated modifications, any bounded neural action either produces a feasible sequence or causes infeasible candidates to be rejected.

Across 24–288 satellite cases and six operating regimes, GRIP-MAPPO improves WOC, TDE, and UTC over greedy, heuristic, vector-state, and homogeneous-graph online baselines while keeping replanning time within a small constant factor of greedy construction. Ablation and sensitivity results further indicate that relation-specific graph encoding, satellite–ground / downlink edges, orbital-group actors, and bounded downlink repair are important for robustness under downlink and emergency stress.

Future work will incorporate higher-fidelity orbital, communication, and energy models to further improve environmental realism. In addition, more interpretable and lightweight representations of the learned state-to-preference mapping, such as policy distillation or surrogate-model approximations, will be investigated to support resource-constrained onboard deployment while preserving the existing scheduler interface.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/rs18132109/s1>, Table S1. Fixed mean-element envelope used by every simulated satellite; Table S2. Scale-dependent constellation layout; Table S3. Ground-station coordinates used in the simulator. Longitudes are east-positive. Heights are WGS-84 ellipsoidal heights when public pages state WGS-84; entries marked as simulation heights are fixed benchmark values used for coordinate conversion; Table S4. Nominal station-rate classes before scenario multipliers; Table S5. Default request distributions before scenario multipliers; Table S6. Nominal operation and resource constants. Scenario multipliers are applied after these defaults; Table S7. Protocol-level definition of the six operating regimes. References [10,51–68] are cited in the Supplementary Materials.

Author Contributions: Conceptualization, F.Z. and G.W. (Gang Wen); methodology, F.Z., X.L. and G.W. (Gang Wen); software, F.Z.; validation, F.Z., T.W. and Z.J.; formal analysis, F.Z., X.L. and J.Z.; investigation, F.Z. and G.W. (Guofang Wang); resources, G.W. (Guofang Wang) and T.W.; data curation, S.H. and J.S.; writing—original draft preparation, F.Z.; writing—review and editing, T.W. and X.L.; visualization, F.Z.; supervision, F.Z., X.L. and T.W.; project administration, G.W. (Gang Wen); funding acquisition, G.W. (Gang Wen) All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Science and Technology Project of Electric Power Research Institute, Yunnan Power Grid Co., Ltd., under Grant No. 056200KC24100059.

Data Availability Statement: The source code and reproducibility package are publicly available at <https://github.com/leecanknown/GRIP-MAPPO> (accessed on 20 June 2026).

Acknowledgments: The authors would like to thank all the reviewers and editors for their comments on this paper.

Conflicts of Interest: Authors Fangrong Zhou, Guofang Wang, Gang Wen, Zonghan Jiao, Jingrun Zhang, Shuangde Huang and Jianwen Sun were employed by the company Yunnan Power Grid Co., Ltd. The authors declare that this study received funding from Yunnan Power Grid Co., Ltd. The funder was not involved in the study design, collection, analysis, interpretation of data, the writing of this article or the decision to submit it for publication.

References

- 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. [\[CrossRef\]](#)
- Ferrari, B.; Cordeau, J.F.; Delorme, M.; Iori, M.; Orosei, R. Satellite scheduling problems: A survey of applications in Earth and outer space observation. *Comput. Oper. Res.* **2025**, *173*, 106875. [\[CrossRef\]](#)
- Liu, R.; Ding, X.; Wu, W.; Guo, W. Distributed Coordination of Space–Ground Multiresources for Remote Sensing Missions. *Remote Sens.* **2023**, *15*, 3362. [\[CrossRef\]](#)
- He, C.; Dong, Y.; Li, H.; Liew, Y. Reasoning-Based Scheduling Method for Agile Earth Observation Satellite with Multi-Subsystem Coupling. *Remote Sens.* **2023**, *15*, 1577. [\[CrossRef\]](#)
- He, C.; Dong, Y. Multi-Satellite Observation-Relay Transmission-Downloading Coupling Scheduling Method. *Remote Sens.* **2023**, *15*, 5639. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- Liu, D.; Zhou, G. Deep Reinforcement Learning-Based Attention Decision Network for Agile Earth Observation Satellite Scheduling. *Remote Sens.* **2024**, *16*, 4436. [\[CrossRef\]](#)
- Gabrel, V.; Vanderpooten, D. Enumeration and interactive selection of efficient paths in a multiple criteria graph for scheduling an Earth observing satellite. *Eur. J. Oper. Res.* **2002**, *139*, 533–542. [\[CrossRef\]](#)
- Vasquez, M.; Hao, J.K. A logic-constrained knapsack formulation and a tabu algorithm for the daily photograph scheduling of an Earth-observation satellite. *Comput. Optim. Appl.* **2001**, *20*, 137–157. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- Bianchessi, S.; Righini, G. Planning and scheduling algorithms for the COSMO-SkyMed constellation. *Aerosp. Sci. Technol.* **2008**, *12*, 535–544. [\[CrossRef\]](#)
- Cho, D.H.; Kim, J.H.; Choi, H.L.; Ahn, J. Optimization-based scheduling method for agile Earth-observing satellite constellation. *J. Aerosp. Inf. Syst.* **2018**, *15*, 611–626. [\[CrossRef\]](#)
- Chen, X.; Reinelt, G.; Dai, G.; Spitz, A. A mixed integer linear programming model for multi-satellite scheduling. *Eur. J. Oper. Res.* **2019**, *275*, 694–707. [\[CrossRef\]](#)
- Valicka, C.G.; Garcia, D.; Staid, A.; Watson, J.P.; Hackebeil, G.; Rathinam, S.; Ntamo, L. Mixed-integer programming models for optimal constellation scheduling given cloud cover uncertainty. *Eur. J. Oper. Res.* **2019**, *275*, 431–445. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- Seman, L.O.; Rigo, C.A.; Camponogara, E.; Munari, P. Improving energy aware nanosatellite task scheduling by a branch-cut-and-price algorithm. *Comput. Oper. Res.* **2023**, *158*, 106292. [\[CrossRef\]](#)
- Antunes, P.M.; Seman, L.O.; Camponogara, E. A branch-and-price algorithm for energy aware task scheduling of constellations of nanosatellites. *Comput. Oper. Res.* **2026**, *185*, 107259. [\[CrossRef\]](#)
- Wang, X.; Gu, Y.; Wu, G.; Woodward, J.R. Robust scheduling for multiple agile Earth observation satellites under cloud coverage uncertainty. *Comput. Ind. Eng.* **2021**, *156*, 107292. [\[CrossRef\]](#)
- Wu, J.; Song, B.; Zhang, G.; Ou, J.; Chen, Y.; Yao, F.; He, L.; Xing, L. A data-driven improved genetic algorithm for agile Earth observation satellite scheduling with time-dependent transition time. *Comput. Ind. Eng.* **2022**, *174*, 108823. [\[CrossRef\]](#)
- Liu, M.; Wu, G.; Gu, Y.; Luo, Q. An Ensemble of Heuristic Adaptive Contract Net Protocol for Efficient Dynamic Data Relay Satellite Scheduling Problem. *Aerospace* **2025**, *12*, 749. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- He, L.; Liang, B.; Li, J.; Sheng, M. Joint observation and transmission scheduling in agile satellite networks. *IEEE Trans. Mob. Comput.* **2022**, *21*, 4381–4396. [\[CrossRef\]](#)
- Hu, X.; Zhu, W.; An, B.; Jin, P.; Xia, W. A branch and price algorithm for EOS constellation imaging and downloading integrated scheduling problem. *Comput. Oper. Res.* **2019**, *104*, 74–89. [\[CrossRef\]](#)
- Zhang, J.; Xing, L. An improved genetic algorithm for the integrated satellite imaging and data transmission scheduling problem. *Comput. Oper. Res.* **2022**, *139*, 105626. [\[CrossRef\]](#)
- Rocha, Y.; Chagas, G.O.; Coelho, L.C.; Subramanian, A. The integrated agile Earth observation satellite scheduling problem. *Comput. Oper. Res.* **2025**, *184*, 107212. [\[CrossRef\]](#)
- Shang, M.; Yuan, R.; Song, B.; Huang, X.; Yang, B.; Li, S. Joint observation and transmission scheduling of multiple agile satellites with energy constraint using improved ACO algorithm. *Acta Astronaut.* **2025**, *230*, 92–103. [\[CrossRef\]](#)

28. Qu, Q.; Liu, K.; Li, X.; Zhou, Y.; Lu, J. Satellite observation and data-transmission scheduling using imitation learning based on mixed integer linear programming. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *59*, 1989–2001. [\[CrossRef\]](#)
29. Wen, Z.; Li, L.; Song, J.; Zhang, S.; Hu, H. Scheduling single-satellite observation and transmission tasks by using hybrid actor–critic reinforcement learning. *Adv. Space Res.* **2023**, *71*, 3883–3896. [\[CrossRef\]](#)
30. Liu, M.; Fan, M.; Gu, Y.; Wang, Y.; Wu, G. Proximal Policy Optimization-Based Hierarchical Distributed Auction for Dual-Layer Inter-Layer Link Scheduling in Remote Sensing. *IEEE Trans. Autom. Sci. Eng.* **2026**, *23*, 6858–6875. [\[CrossRef\]](#)
31. Cheng, J.; Fan, M.; Gu, Y.; Tang, W.; Luo, Q.; Wang, Y.; Wang, X.; Wu, G. UHTS-DRL: A deep reinforcement learning framework for integrated agile satellite observation and data transmission scheduling. *Inf. Sci.* **2026**, *740*, 123200. [\[CrossRef\]](#)
32. Vinyals, O.; Fortunato, M.; Jaitly, N. Pointer networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2015; pp. 2692–2700.
33. Kool, W.; van Hoof, H.; Welling, M. Attention, learn to solve routing problems! In *Proceedings of the International Conference on Learning Representations*, New Orleans, LA, USA, 6–9 May 2019.
34. 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. [\[CrossRef\]](#)
35. Wei, L.; Chen, Y.; Chen, M.; Chen, Y. Deep reinforcement learning and parameter transfer based approach for the multi-objective agile Earth observation satellite scheduling problem. *Appl. Soft Comput.* **2021**, *110*, 107607. [\[CrossRef\]](#)
36. Li, D.; Wang, H.; Yang, Z.; Gu, Y.; Shi, S. An online distributed satellite cooperative observation scheduling algorithm based on multiagent deep reinforcement learning. *IEEE Geosci. Remote. Sens. Lett.* **2021**, *18*, 1901–1905. [\[CrossRef\]](#)
37. Wang, X.; Wu, J.; Shi, Z.; Zhao, F.; Jin, Z. Deep reinforcement learning-based autonomous mission planning method for high and low orbit multiple agile Earth observing satellites. *Adv. Space Res.* **2022**, *70*, 3478–3493. [\[CrossRef\]](#)
38. Wang, X.; Zhao, F.; Shi, Z.; Jin, Z. Deep reinforcement learning-based periodic Earth observation scheduling for agile satellite constellation. *J. Aerosp. Inf. Syst.* **2023**, *20*, 508–519. [\[CrossRef\]](#)
39. Ou, J.; Xing, L.; Yao, F.; Li, M.; Lv, J.; He, Y.; Song, Y.; Wu, J.; Zhang, G. Deep reinforcement learning method for satellite range scheduling problem. *Swarm Evol. Comput.* **2023**, *77*, 101233. [\[CrossRef\]](#)
40. Long, Y.; Shan, C.; Shang, W.; Li, J.; Wang, Y. Deep reinforcement learning-based approach with varying-scale generalization for the Earth observation satellite scheduling problem considering resource consumptions and supplements. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 2572–2585. [\[CrossRef\]](#)
41. Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 6379–6390.
42. Yu, C.; Velu, A.; Vinitzky, E.; Gao, J.; Wang, Y.; Bayen, A.; Wu, Y. The surprising effectiveness of MAPPO in cooperative, multi-agent games. In *Proceedings of the Advances in Neural Information Processing Systems*, New Orleans, LA, USA, 28 November–9 December 2022.
43. Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; Yu, P.S. Heterogeneous graph attention network. In *Proceedings of the World Wide Web Conference*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 2022–2032.
44. 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. [\[CrossRef\]](#)
45. Chen, M.; Du, Y.; Tang, K.; Xing, L.; Chen, Y.; Chen, Y. Learning to construct a solution for the agile satellite scheduling problem with time-dependent transition times. *IEEE Trans. Syst. Man Cybern. Syst.* **2024**, *54*, 5949–5963. [\[CrossRef\]](#)
46. Khalil, E.B.; Bodic, P.L.; Song, L.; Nemhauser, G.; Dilkina, B. Learning to branch in mixed integer programming. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI Press: Washington, DC, USA, 2016.
47. Gasse, M.; Chételat, D.; Ferroni, N.; Charlin, L.; Lodi, A. Exact combinatorial optimization with graph convolutional neural networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2019.
48. Hottung, A.; Tierney, K. Neural large neighborhood search for the capacitated vehicle routing problem. In *Proceedings of the 24th European Conference on Artificial Intelligence*, Santiago de Compostela, Spain, 29 August–8 September 2020. [\[CrossRef\]](#)
49. Zhang, J.; Liu, C.; Zhao, W.; Xu, K. Learning to dispatch for job shop scheduling via deep reinforcement learning. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2020.
50. Huang, W.; Li, Z.; He, X.; Xiang, J.; Du, X.; Liang, X. DRL-Based Dynamic Destroy Approaches for Agile-Satellite Mission Planning. *Remote Sens.* **2023**, *15*, 4503. [\[CrossRef\]](#)
51. CelesTrak. Planet GP/TLE Data Group. Available online: <https://celestrak.org/norad/elements/table.php?FORMAT=tle&GROUP=planet> (accessed on 30 April 2026).
52. CelesTrak. NORAD General Perturbations Element Sets: Current Data. Available online: <https://www.celestrak.org/NORAD/elements/> (accessed on 30 April 2026).
53. Vallado, D.A.; Crawford, P.; Hujsak, R.; Kelso, T.S. Revisiting Spacetrack Report #3. In *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference and Exhibit*; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2006.

54. NASA Navigation and Ancillary Information Facility; ESA SPICE Service. ESTRACK Topocentric Frame Kernel, Version 0.3. Available online: https://naif.jpl.nasa.gov/pub/naif/pds/wgc/kernels/fk/estrack_v03.tf (accessed on 30 April 2026).
55. European Space Agency. Estrack: ESA's Global Ground Station Network. Available online: https://www.esa.int/Enabling_Support/Operations/ESA_Ground_Stations/Estrack_ESA_s_global_ground_station_network (accessed on 30 April 2026).
56. European Space Agency. Kiruna Station. Available online: https://www.esa.int/Enabling_Support/Operations/ESA_Ground_Stations/Kiruna_station (accessed on 30 April 2026).
57. European Space Agency. Kourou Station. Available online: https://www.esa.int/Enabling_Support/Operations/ESA_Ground_Stations/Kourou_station (accessed on 30 April 2026).
58. European Space Agency. Maspalomas Station. Available online: https://www.esa.int/Enabling_Support/Operations/ESA_Ground_Stations/Maspalomas_station (accessed on 30 April 2026).
59. European Space Agency. Perth Station. Available online: https://www.esa.int/Enabling_Support/Operations/ESA_Ground_Stations/Perth_station (accessed on 30 April 2026).
60. European Space Agency. Redu Facilities. Available online: https://www.esa.int/ESA_Multimedia/Images/2007/08/Redu_facilities2 (accessed on 30 April 2026).
61. European Space Agency. Santa Maria Station. Available online: https://www.esa.int/Enabling_Support/Operations/ESA_Ground_Stations/Santa_Maria_station (accessed on 30 April 2026).
62. European Space Agency. Villafranca Station. Available online: https://www.esa.int/Enabling_Support/Operations/ESA_Ground_Stations/Villafranca_station (accessed on 30 April 2026).
63. European Space Agency. ESA Ground Stations Live. Available online: https://www.esa.int/Enabling_Support/Operations/ESA_Ground_Stations/ESA_Ground_Stations_Live (accessed on 30 April 2026).
64. European Space Agency. Santiago Station. Available online: https://www.esa.int/Enabling_Support/Operations/Santiago_station (accessed on 30 April 2026).
65. European Space Agency. Galileo IOV Ground Stations: Svalbard. Available online: https://www.esa.int/Applications/Satellite_navigation/Galileo/Galileo_IOV_ground_stations_Svalbard (accessed on 30 April 2026).
66. Chen, J.; Chen, M.; Wen, J.; He, L.; Liu, X. A Heuristic Construction Neural Network Method for the Time-Dependent Agile Earth Observation Satellite Scheduling Problem. *Mathematics* **2022**, *10*, 3498. [CrossRef]
67. 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]
68. Antuori, V.; Wojtowicz, D.T.; Hebrard, E. Solving the Agile Earth Observation Satellite Scheduling Problem with CP and Local Search. In Proceedings of the 31st International Conference on Principles and Practice of Constraint Programming (CP 2025), Glasgow, Scotland, 10–15 August 2025; Volume 340, pp. 3:1–3:22. [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.