AstroReason-Bench: Evaluating Unified Agentic Planning across Heterogeneous Space Planning Problems

Anonymous ACL submission

Abstract

Recent advances in agentic Large Language Models (LLMs) have positioned them as generalist planners capable of reasoning and acting across diverse tasks. However, existing agent benchmarks largely focus on symbolic or weakly grounded environments, leaving their performance in physics-constrained real-world domains underexplored. We introduce *AstroReason-Bench*, a comprehensive benchmark for evaluating agentic planning in *Space Planning Problems (SPP)*, a family of high-stakes problems with heterogeneous objectives, strict physical constraints, and long-horizon decision-making. AstroReason-Bench integrates multiple scheduling regimes, including ground station communication and agile Earth observation, and provides a unified agent-oriented interaction protocol. Evaluating on a range of state-of-the-art open- and closed-source agentic LLM systems, we find that current agents substantially underperform specialized solvers, highlighting key limitations of generalist planning under realistic constraints. AstroReason-Bench offers a challenging and diagnostic testbed for future agentic research.

1 Introduction

Recent progress in large language models has given rise to *agentic systems* that integrate natural language reasoning with planning, tool use, and iterative decision-making. These systems are increasingly viewed as *generalist planners*, capable of addressing diverse tasks without task-specific algorithm design, ranging from software engineering and web automation to scientific reasoning and decision support.

Despite these advances, the evaluation of agentic systems remains limited. Existing benchmarks primarily focus on symbolic, text-based, or weakly grounded environments—such as web navigation, code synthesis, or synthetic games (Zhou et al.; Jimenez et al.; Paglieri et al.). While valuable for assessing reasoning and tool orchestration, these settings abstract away hard physical constraints, long-horizon planning requirements, and irreversible feasibility boundaries. Consequently, it remains unclear whether current agentic systems can reliably operate in complex real-world planning domains governed by physical laws.

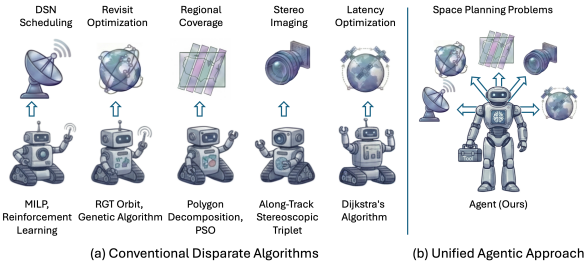

Figure 1: Transition from disparate algorithms to a unified agentic framework: (a) illustrates the conventional methodology where tasks are isolated and optimized using disparate algorithms; (b) presents our unified agentic system, where a central intelligent agent leverages a toolkit to manage disparate scheduling tasks in an integrated manner.

Space Planning Problems (SPP) offer a uniquely challenging and underexplored testbed for generalist planning. SPP encompass heterogeneous objectives, strict physical and temporal constraints, large combinatorial action spaces, and long-horizon decision-making. These challenges arise across structurally distinct sub-problems, including ground station communication scheduling, agile Earth observation planning, and deep-space network allocation. Historically, each of these problems has been tackled using highly specialized optimization techniques, such as mixed-integer programming (Guillaume et al., 2007; Claudet et al., 2022), heuristic search (Milena et al., 2025; Zezhong et al., 2023), or reinforcement learning (Herrmann and Schaub, 2023; Li and Wang, 2025; Lyu et al., 2024).

While benchmarks and simulators exist for indi-

vidual SPP sub-problems, they are typically developed in isolation, with incompatible assumptions, interfaces, and evaluation metrics. As a result, they are well-suited for assessing specialized solvers but ill-suited for evaluating whether a single *agentic* system can adapt its reasoning and tool usage across multiple, structurally diverse planning environments.

To address this gap, we introduce **AstroReason-Bench**, a comprehensive, physics-aligned benchmark suite for evaluating agentic planning in SPP. AstroReason-Bench integrates multiple representative SPP sub-problems under a unified, agent-oriented interaction and evaluation protocol, treating them as a family of heterogeneous environments that collectively stress-test the adaptability and robustness of generalist planners.

We evaluate AstroReason-Bench using a range of state-of-the-art open- and closed-source agentic LLM systems, including DeepSeek V3.2, Claude Sonnet 4.5, Gemini 3 Flash, etc. To enable zero-shot operation, we provide a minimal set of task-relevant tools via the Model Context Protocol (MCP), allowing agents to observe environment states, invoke simulators, and execute scheduling decisions.

Our empirical results reveal a substantial performance gap between current agentic systems and specialized optimization methods, highlighting the challenges posed by strict physical constraints. We argue that this gap underscores the realism and diagnostic value of AstroReason-Bench, which serves both as a rigorous evaluation platform and as a foundation for future research in agentic planning, transfer, and learning for space planning problems.

Our contributions are summarized as follows:

- We introduce AstroReason-Bench, the first unified benchmark suite for evaluating agentic planning across diverse space planning problems.
- We provide standardized, agent-oriented interfaces and metrics enabling consistent evaluation across heterogeneous SPP tasks.
- We present a comprehensive evaluation of state-of-the-art agentic LLM systems, revealing key limitations and open challenges in physics-grounded planning.

2 Related Works

2.1 The Landscape of Satellite Planning and Scheduling

Satellite scheduling is characterized by fragmented, domain-specific optimization paradigms. **DSN Scheduling**, dealing with antenna oversubscription, has progressed from heuristic repair (Johnston et al., 2009; Johnston and Clement, 2006) to MILP (Guillaume et al., 2007) and RL-based benchmarks like SatNet (Goh et al., 2021). **Earth Observation** involves complex kinematic constraints. Agile satellites require specialized heuristics (e.g., ALNS, PSO) for stereoscopic imaging (Zezhong et al., 2023; Bagnardi et al., 2016; Lemaître et al., 2002) and polygon decomposition for large-area coverage (Li, 2017; Milena et al., 2025; Hu et al., 2021). Similarly, constellation-level monitoring often relies on tailored repeat ground tracks (Lee et al., 2024; Li and Wang, 2025). **Integrated Sensing and Communication (ISAC)** adds real-time routing challenges, often addressed via Multi-Agent RL (Lyu et al., 2024; Wu et al., 2025; Cao et al., 2022). This fragmentation necessitates a unified interface that can adapt across these heterogeneous domains.

2.2 Agentic Planning and Reasoning

LLMs are evolving from static models to agentic planners capable of tool use and reasoning (Kojima et al., 2022; Wang et al.; Wei et al., 2025). While benchmarks like PlanBench (Valmeekam et al., 2023a) and TravelPlanner (Xie et al., 2024) evaluate symbolic reasoning, they often lack the high-fidelity physical constraints of engineering domains. Recent interactive agent benchmarks (e.g., τ-bench (Yao et al., 2024)) further evaluate tool use and execution feedback, but similarly abstract away domain-specific physical dynamics. Agents offer a promising universal interface for physical systems, acting as "co-pilots" that translate natural language into executable plans or API calls (Liang et al.; Li et al., 2025; Valmeekam et al., 2023b). Unlike rigid specialized solvers, agentic systems can potentially handle nuanced constraints zero-shot. This work benchmarks this capability within the rigorous constraints of space mission planning.

3 The AstroReason-Bench Suite

We introduce AstroReason-Bench, a comprehensive evaluation suite designed to evaluate autonomous agents under high-fidelity orbital, re-

source and temporal constraints. It integrates the legacy SatNet environment (Goh et al., 2021) with four novel, procedurally generated mission profiles.

3.1 Simulation Environment & Constraints

The engine uses the Simplified General Perturbations 4 (SGP4) model (Hoots and Roehrich, 1980; Vallado et al.), a standard analytical propagator for consistency with real-world Two-Line Element (TLE) data, a standardized format for encoding the orbital elements of Earth-orbiting objects (Vallado et al.). The simulation enforces three primary constraint classes:

Resource Constraints Agents must manage two coupled resource buffers.

- **Energy ($E(t)$):** Modeled as an integral of power generation P_{gen} (solar) minus power consumption P_{con}. P_{gen} is conditional on the satellite's eclipse status (computed via conical shadow projection). The constraint requires $E(t) = E(0) + \int_0^t (P_{gen}(t) - P_{con}(t)) \geq 0, \forall t$.
- **Data Storage ($D(t)$):** Modeled as a buffer with inflow from observations and outflow from downlinks. Agents must schedule ground station passes to prevent buffer overflows ($D(t) \leq D_{max}$) where D_{max} is the maximum onboard storage of a satellite.

Kinematic Constraints For Earth observation tasks, satellites are modeled as agile bodies requiring attitude maneuvers. A maneuver between target i and target j is valid only if the temporal gap Δt_{ij} satisfies $\Delta t_{ij} \geq t_{slew} + t_{settle}$. While the settling time t_{settle} is modeled as a constant, the slew time t_{slew} is derived from a trapezoidal velocity profile based on the angular displacement $\Delta\theta_{ij} = 2\arccos|\mathbf{q}_i \cdot \mathbf{q}_j|$, where $\mathbf{q}$ denotes the unit quaternion. Given maximum angular velocity ω_{max} and acceleration α_{max}, t_{slew} is defined as:

$$t_{slew} = \begin{cases} 2\sqrt{\frac{\Delta\theta_{ij}}{\alpha_{max}}} & \text{if } \Delta\theta_{ij} < \frac{\omega_{max}^2}{\alpha_{max}} \\ \frac{\Delta\theta_{ij}}{\omega_{max}} + \frac{\omega_{max}}{\alpha_{max}} & \text{otherwise} \end{cases} \quad (1)$$

Concurrency Constraints In contrast, link terminals (Downlink/Inter-Satellite Link) are gimbaled and rotationally independent. They do not induce attitude constraints and can operate concurrently with observations. Link validity is checked solely against terminal capacity N_{term} (maximum simultaneous links) and resource budgets, ignoring slew dynamics.

3.2 Benchmark Tasks

AstroReason-Bench unifies five distinct planning challenges. While the first is an adaptation of an existing standard, the latter four are novel contributions generated procedurally.

Benchmark 1: SatNet (DSN Scheduling) We incorporate the SatNet environment (Goh et al., 2021), a standard benchmark for Deep Space Network (DSN) scheduling. The objective is to minimize the unsatisfied time of resource allocation across competing requests. Using the original metrics, we define the unsatisfied ratio for mission $m \in \mathcal{M}$ as $U_m = (T^m_{req} - T^m_{alloc})/T^m_{req}$, where T^m_{req} and T^m_{alloc} are the total requested and allocated durations for mission m, and $\mathcal{M}$ is the set of missions. The primary metrics are the RMS unsatisfied ratio $U_{rms} = \sqrt{\frac{1}{\mathcal{M}}\sum_{m\in\mathcal{M}}(U_m)^2}$ and the max unsatisfied ratio $U_{max} = \max_{m\in\mathcal{M}} U_m$.

Benchmark 2: Revisit Optimization

- **Monitoring Targets:** Let $\mathcal{T}_{mon}$ be the set of targets requiring continuous observation. We minimize the *Revisit Gap*, defined as the time interval between consecutive observations. Let Δi be the set of gaps for target i. The primary metric is the global average gap:

$$M_{gap} = \frac{1}{|\mathcal{T}_{mon}|}\sum_{i\in\mathcal{T}_{mon}} \text{mean}(\Delta_i) \quad (2)$$

- **Mapping Targets:** Require a fixed quota of observations. Success is measured by the *Coverage Ratio* (M_{map}), the percentage of quotas fulfilled.

Benchmark 3: Regional Coverage Designed for satellites capable of strip-imaging modes, such as SKYSAT[1] and ICEYE[2], this task requires maximizing the area covered within polygons. Unlike point targets, this requires the agent to plan continuous swaths to maximize the coverage of complex polygonal regions. Let $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$ be the set of non-overlapping target polygons, and $\mathcal{S} = \bigcup_j S_j$ represent the union of all scheduled observation strips S_j. The coverage performance is evaluated using Area-based Recall (AR), defined

[1] https://earth.esa.int/eogateway/missions/skysat

[2] https://www.iceye.com/

as the ratio of the captured target area to the total required area:

$$M_{cov} = \frac{\text{Area}\left(\mathcal{S} \cap \left(\bigcup_{p \in \mathcal{P}} p\right)\right)}{\sum_{p \in \mathcal{P}} \text{Area}(p)} \tag{3}$$

Benchmark 4: Stereo Imaging This task simulates high-value missions requiring 3D reconstruction. Unlike standard acquisitions, a stereo product is only valid if a target is captured as a doublet of observations that satisfies strict geometric and temporal synchronization. These constraints ensure sufficient parallax for depth estimation while minimizing radiometric changes between images. A doublet is valid if it satisfies the following system:

$$\begin{cases} \Delta\theta_{az}^{min} \leq |\theta_{az,1} - \theta_{az,2}| \leq \Delta\theta_{az}^{max} \\ |t_1 - t_2| \leq T_{max} \\ \min(\theta_{el,1}, \theta_{el,2}) \geq \theta_{el}^{min} \end{cases} \tag{4}$$

where θ_{az} and θ_{el} represent the azimuth and elevation angles, respectively. The constraint on $|\Delta\theta_{az}|$ and θ_{el} ensure an appropriate geometric baseline for stereo reconstruction. Specifically, in multi-pass scenarios, the temporal component of azimuth separation serves as a determinant for metadata error correlation; accounting for this correlation is essential for accurate vertical error prediction (Dolloff and Theiss, 2012).

Benchmark 5: Latency-Optimization This task models a Low Earth Orbit (LEO) mega-constellation providing Integrated Sensing and Communications (ISAC) services, such as QIANFAN[3]. The agent must manage the inherent resource contention between high-priority communication links and opportunistic Earth observation.

- **Communication Services:** The objective is to maintain persistent connectivity between ground-station pairs. Performance is quantified by Availability (M_{avail}), the fraction of time steps where at least one valid routing path exists, and Mean Latency (M_{lat}). We define M_{lat} as the time-averaged propagation delay of the shortest path available at each epoch:

$$M_{lat} = \frac{1}{\mathcal{T}_{valid}} \sum_{t \in \mathcal{T}_{valid}} \min_{p \in \mathcal{P}_t} \text{delay}(p) \tag{5}$$

where $\mathcal{P}_t$ is the set of all feasible paths at time t, and $\mathcal{T}_{valid}$ denotes the set of time steps with non-zero availability.

- **Opportunistic Mapping:** Simultaneously, the fleet must fulfill a fixed observation quota for mapping targets $\mathcal{T}_{map}$, as defined in Benchmark 2. This requires the agent to exploit idle time-frequency resources or satellite overflights that do not compromise the primary communication backhaul. The metric is the Coverage Ratio (M_{map}), representing the percentage of completed quotas.

3.3 Procedural Dataset Generation

The generation process ensures diversity and physical validity.

- **Constellation Sampling:** We sample specific constellation archetypes (e.g., QIANFAN for communications, mixtures of SPOT/PLEIADES for stereo imaging) to preserve realistic orbital distributions. From these families, we subsample 10 to 100 satellites using archived TLE data.

- **Target Distribution:** Ground targets are sampled from a global database of 40,000+ cities. To ensure feasibility, targets are dynamically filtered based on the *average inclination* of the selected constellation, ensuring they fall within accessible latitude bands.

- **Temporal Horizon:** All generated scenarios span a fixed 4-day planning horizon (2025-07-17T12:00:00 to 2025-07-21T12:00:00). This interval was chosen to align with the epoch of our TLE dataset, minimizing propagation errors while providing a sufficiently long horizon to test long-term resource management and periodic revisit patterns.

- **Problem Scaling:** We control difficulty by maintaining specific *Resource-to-Request* ratios. For example, Revisit Optimization typically maintains about a 4:1 satellite-to-target ratio, whereas Stereo Imaging enforces about a tighter 1:1 ratio to induce high resource contention.

All generated scenarios are serialized into a standard JSON/YAML format, ensuring that the benchmark is reproducible and model-agnostic.

[3] https://en.wikipedia.org/wiki/Qianfan

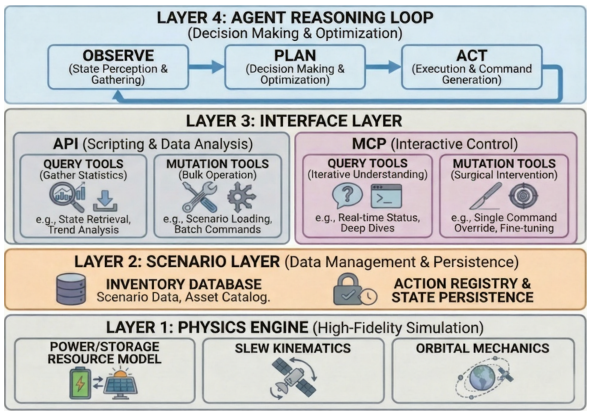

Figure 2: **The Environment and Interface Architecture.** The architecture is organized into four layers: (1) The Physics Layer handles stateless physics computation; (2) The Scenario Layer manages session state; (3) The Interface Layer provides access to the environment via semantic MCP tools and a Python API; and (4) The Cognitive Layer hosts the LLM agent.

4 Environment and Interface Design

Existing benchmarks for agentic software engineering rely on standard compilers and interpreters (e.g., GCC, Python) as their execution environment. In the domain of space planning, while high-fidelity simulators exist (e.g., STK[4], Basilisk (Kenneally et al., 2020)), they are primarily designed for human experts via GUIs or complex scripting environments, lacking standardized interfaces accessible to autonomous agents. AstroReason-Bench addresses this by establishing a system architecture that wraps physics models into agent-ready tools.

4.1 Layer 1: Physics Engine (Stateless)

This layer serves as the immutable "laws of physics" for the environment, integrating three core models: (1) **SGP4 Propagation**: high-precision orbital propagation provides ground truth for satellite states and geometric visibility; (2) **Slew Kinematics**: a trapezoidal velocity model simulates slew maneuvers for agile satellites, enforcing settling time constraints; and (3) **Resource Modeling**: a resource event manager models power generation (solar) and consumption (action), while handling storage inflow/outflow dynamics for observation and downlink activities.

4.2 Layer 2: Scenario Manager (Stateful)

This layer acts as the session controller, maintaining the scenario state. It manages three critical components: (1) **Inventory Database**: a read-only registry of satellites, targets, and stations loaded from external catalogs; (2) **Action Registry**: a mutable timeline tracking all staged actions validating against the mission schema; and (3) **State Persistence**: a file-backed mechanism guarded by advisory locks. To ensure consistency across both interfaces in Layer 3, this locking mechanism enforces atomic updates, preventing race conditions between the semantic and programmatic modalities.

4.3 Layer 3: Interface Abstraction

This layer provides the critical bridge between the agent and the physics kernel, exposing the environment through the two complementary modalities: **(1) Semantic MCP:** the MCP is designed for exploration and interactive debugging. It exposes the environment state as human-readable JSON summaries optimized for the LLM's context window. Key capabilities include state inspection, action staging/unstaging, and rich semantic feedback on constraint violations; **(2) Programmatic Python API:** to address the arithmetic limitations of LLMs, we expose a Python API distributed as a local repository. This allows agents to write and execute scripts for batch computation and custom heuristic implementation.

4.4 Layer 4: Cognitive Layer

This layer represents the agent under evaluation. We employ a standard ReAct (Yao et al.) loop via Claude Code[5] as the foundation, where the LLM maintains a high-level mission plan and interacts with the lower layers to refine and validate its strategy.

5 Experiments

We evaluate a range of state-of-the-art LLM-based agentic systems on the AstroReason-Bench suite along two dimensions: (1) quantitative benchmarking against traditional optimization baselines, and (2) qualitative case studies analyzing the reasoning behaviors of agentic workflows (Section A.2).

5.1 Experiment Setup

We conducted large-scale evaluation evolving 150 full mission simulations across five benchmark categories. Each simulation involves an LLM agent

[4] https://www.ansys.com/products/missions/ansys-stk

[5] https://docs.anthropic.com/en/docs/agents-and-tools/claude-code

operating autonomously within a sandboxed environment, querying orbital mechanics APIs, staging actions, and committing final plans subject to physical validation.

Models Our model suite includes six frontier LLM agents: **Claude Sonnet 4.5**, **Gemini 3 Flash**, **DeepSeek V3.2** (Liu et al., 2025), **Qwen3 Coder** (Yang et al., 2025), **DeepSeek V3.1 Nex N1** (Cai et al., 2025) and **Kat Coder Pro** (Zhan et al., 2025). Each model completed 5 cases per benchmark (25 runs per model), with a 2-hour timeout per case. Computation was restricted to 16GB memory and 8 CPU cores (AMD Ryzen 7 9700X), representing a constrained but realistic deployment scenario.

Baselines For SatNet, we compare against four published baselines: (1) **Unweighted** and (2) **Randomized**, two greedy heuristics that schedule activities in order of duration or randomly, proposed by Guillaume et al. (Guillaume et al., 2007); (3) **Δ-MILP**, a Mixed-Integer Linear Programming solver (Claudet et al., 2022); and (4) **RL (PPO)**, a reinforcement learning approach trained via Proximal Policy Optimization (Goh et al., 2021). These results are cited from the respective publications. For our novel benchmarks (Revisit Optimization, Regional Coverage, Latency Optimization, Stereo Imaging), we implement two traditional algorithms:

- **Greedy Heuristics**: a domain-aware greedy scheduler that scores candidate windows using benchmark-specific heuristics (e.g., gap-since-last-observation for Revisit Optimization, azimuth separation for Stereo Imaging) and stages the highest-scoring valid action at each step.
- **Simulated Annealing (SA)**: a metaheuristic that represents solutions as binary masks over candidate windows, uses neighbor generation (add/remove/swap operations), and accepts worse solutions probabilistically via the Metropolis criterion to escape local minima.

5.2 Main Results

5.2.1 Benchmark 1: SatNet (Deep Space Network Scheduling)

On SatNet, all LLM agents achieve U_{rms} scores between 0.53–0.59, substantially improving over unweighted/randomized baselines (∼0.87–0.89) but falling short of specialized approaches. The

Method	$\mathbf{U_{max}} \downarrow$	$\mathbf{U_{rms}} \downarrow$
Unweighted (Guillaume et al., 2007)	*1.00*	*0.87*
Randomized (Guillaume et al., 2007)	*1.00*	*0.89*
Δ-MILP (Claudet et al., 2022)	*0.67*	*0.30*
RL (PPO) (Goh et al., 2021)	*0.77*	*0.32*
Claude Sonnet 4.5	1.00	0.55
Gemini 3 Flash	1.00	0.53
DeepSeek V3.2	1.00	0.57
Qwen3 Coder	1.00	0.56
DeepSeek V3.1 Nex N1	1.00	0.58
Kat Coder Pro	1.00	0.59

Table 1: **SatNet Results.** U_{max}: maximum unsatisfied ratio (lower is better); U_{rms}: RMS unsatisfied ratio (lower is better). LLM agents outperform simple heuristics but lag behind specialized optimizers (MILP, RL).

Δ-MILP solver achieves $U_{rms} = 0.30$ through exhaustive combinatorial optimization, while RL (PPO) reaches 0.32 via thousands of training episodes. LLM agents, operating zero-shot without domain-specific training, demonstrate reasonable scheduling intuition but lack the systematic search capabilities of purpose-built optimizers.

5.2.2 Benchmark 2: Revisit Optimization

Method	$\mathbf{M_{map}} \uparrow$	$\mathbf{M_{gap}}$(h) $\downarrow$
Greedy Heuristic	0.32	42.27
SA	1.00	13.65
Claude Sonnet 4.5	1.00	18.83
Gemini 3 Flash	0.86	24.96
DeepSeek V3.2	0.64	29.89
Qwen3 Coder	0.29	38.58
DeepSeek V3.1 Nex N1	0.61	26.78
Kat Coder Pro	0.88	22.46

Table 2: **Revisit Optimization Results.** M_{map}: average mapping target coverage ratio (higher is better); M_{gap}: average mean revisit gap in hours (lower is better). SA outperforms all agents.

Analysis SA achieves the best overall performance ($M_{gap} = 13.65$h) by iteratively optimizing a fitness function that directly measures gap statistics. Among LLM agents, Claude Sonnet 4.5 leads with $M_{gap} = 18.83$h, demonstrating effective gap-aware scheduling while maintaining full mapping coverage.

The Greedy baseline's poor mapping coverage (M_{map}=0.32) reveals a critical failure mode: its heuristic assigns low priority to downlink windows relative to observations, causing satellites to exhaust onboard storage before completing required observations. This illustrates how nearsighted

scheduling without resource lifecycle awareness leads to cascading constraint violations.

Weaker agents (Qwen3 Coder at $M_{gap} = 0.29$) exhibit similar storage management failures, suggesting that resource planning (balancing data acquisition against downlink capacity) is a key differentiator among LLM agents.

5.2.3 Benchmark 3: Regional Coverage

Method	$\mathbf{M_{cov}} \uparrow$
Greedy Heuristic	0.00
SA	0.03
Claude Sonnet 4.5	0.00
Gemini 3 Flash	0.11
DeepSeek V3.2	0.05
Qwen3 Coder	0.03
DeepSeek V3.1 Nex N1	0.06
Kat Coder Pro	0.03

Table 3: **Regional Coverage Results.** M_{cov} : mean polygon coverage ratio (higher is better). All methods achieve low coverage.

Analysis Regional coverage proves challenging for all approaches, with even the best agent (Gemini 3 Flash) achieving only 11% coverage. This benchmark requires a fundamentally different strategy: instead of scheduling point observations, agents must decompose polygons into strips (continuous swaths) according to satellite ground tracks before scheduling observations. We identify two primary failure modes:

1. **Strip orientation mismatch**: Agents typically register strips blindly at mission start without querying satellite ground tracks to understanding constellation geometry. Strips perpendicular to satellite velocity vectors yield near-zero valid observation windows.

2. **Storage exhaustion**: Strip observations consume substantial storage. Agents that fail to schedule sufficient downlinks cannot complete planned acquisitions.

5.2.4 Benchmark 4: Stereo Imaging

Method	$\mathbf{M_{cov}} \uparrow$
Greedy Heuristic	0.00
SA	0.00
Claude Sonnet 4.5	0.05
Gemini 3 Flash	0.06
DeepSeek V3.2	0.12
Qwen3 Coder	0.18
DeepSeek V3.1 Nex N1	0.03
Kat Coder Pro	0.06

Table 4: **Stereo Imaging Results.** M_{cov}: stereo pair coverage ratio (higher is better). Baselines completely fail; LLM agents achieve modest success through constraint-aware scheduling.

Analysis Both baselines achieve 0% stereo coverage, while LLM agents reach up to 18% (Qwen3 Coder). This significant performance gap highlights the agents' superior ability to handle compound constraints. The greedy heuristics fail because they optimize for single attributes without "looking ahead" to satisfy the coupled requirement of a second, geometrically distinct observation. In contrast, successful agents explicitly reasoned about the request as a "stereo pair." They utilized the API interface with Python scripts to search for temporal doublets that satisfied all constraints and then staged both actions simultaneously. This capability to reason about interdependent actions represents a key advantage of the agentic paradigm over simple constructive heuristics.

5.2.5 Benchmark 5: Latency Optimization

Method	$\mathbf{M_{map}} \uparrow$	$\mathbf{M_{avail}} \uparrow$	$\mathbf{M_{lat}}$(ms) $\downarrow$
Greedy Heuristic	0.01	0.00	/
SA	0.30	0.00	/
Claude Sonnet 4.5	0.58	0.00	/
Gemini 3 Flash	0.20	0.00	/
DeepSeek V3.2	0.14	0.00	/
Qwen3 Coder	0.48	0.00	/
DeepSeek V3.1 Nex N1	0.09	0.00	/
Kat Coder Pro	0.18	0.07	58.4

Table 5: **Latency Optimization Results.** M_{map}: average mapping target coverage ratio; M_{avail}: average availability; M_{lat}: mean latency in milliseconds. Only Kat Coder Pro establishes any valid inter-station connections.

Analysis Latency optimization is the most demanding benchmark, requiring agents to establish real-time, multi-hop relay chains between geographically distant ground stations. This is not store-and-forward; the entire chain station A $\leftrightarrow$ satellite A $\leftrightarrow$ satellite B $\leftrightarrow$ station B must be active simultaneously.

As shown in Table 5, nearly all agents fail completely on connection coverage ($M_{com} = 0$). Analysis of agent traces reveal a common misconception: agents attempt to find a single satellite visi-

ble to both stations simultaneously, ignoring that Earth's curvature and station separation make this geometrically impossible.

Kat Coder Pro is the sole exception, achieving $M_{com} = 0.07$ with $M_{lat} = 58.4$ ms. This agent correctly recognized that inter-continental links require multi-hop ISL routing and scheduled coordinated satellite-to-satellite handoffs successfully in two out of five cases.

5.2.6 Summary of Findings

Benchmark	Best Baseline	Best Agent	Key Differentiator
SatNet	MILP (0.30)	Gemini (0.53)	Systematic search
Revisit	SA (13.65h)	Claude (18.83h)	Resource lifecycle
Regional	SA (3%)	Gemini (11%)	Orbital geometry
Stereo	—	Qwen3 (18%)	Compound constraints
Latency	—	Kat-Coder (7%)	Network topology

Table 6: **Capability Summary.** Each benchmark isolates a distinct planning competency.

Table 6 reveals a clear pattern. On benchmarks requiring exhaustive combinatorial search (SatNet, Revisit Optimization), specialized solvers dominate; agents lack the systematic exploration needed to compete. Conversely, on benchmarks where baselines completely fail (Stereo Imaging, Latency Optimization), agents achieve modest but non-trivial success by reasoning about compound constraints and network topology. This suggests that the agentic paradigm's strength lies not in raw optimization power, but in its capacity to recognize and adapt to novel problem structures zero-shot.

6 Case Study

To understand why agents succeed or fail, we present a qualitative analysis of agent traces. The following case study, selected from the Latency Optimization benchmark, illustrates a critical failure mode: the inability to reason about physical impossibility and pivot to alternative strategies.

Phenomenon In **Latency Optimization**, where agents control 90 satellites from the QIANFAN constellation, nearly all agents (except Kat Coder Pro) achieved 0% connection coverage. Trace analysis revealed a consistent misconception: agents attempted to establish communication by finding a single satellite simultaneously visible to both ground stations, which is geometrically impossible in most scenarios due to Earth's curvature and LEO orbital altitudes.

Example A failing agent (e.g., DeepSeek V3.2) queried satellites' access windows to both stations, and when this returned no common windows, the agent concluded the task was infeasible rather than considering multi-hop relay chains.

Contrast One of Kat Coder Pro's successful runs explicitly computed inter-satellite link (ISL) windows and staged an "ISL backbone" between "QIANFAN-1", "QIANFAN-7" and "QIANFAN-10", enabling end-to-end connectivity, at least to a minimal extent. This conceptual leap from seeking a common view to constructing a network path is illustrated in Figure 3.

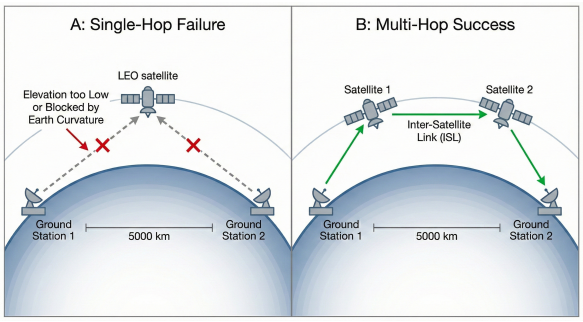

Figure 3: **Geometry of Long-Range ISAC.** Single-satellite visibility (left) is often impossible for distant ground stations. Connectivity requires multi-hop routing via Inter-Satellite Links (right), a spatial reasoning step most agents missed.

Implication Agents struggle to recognize the geometrical or physical infeasibility of naive solutions and therefore fail to pivot toward alternatives. This suggests deficits in spatial reasoning ability.

7 Conclusion

We introduced **AstroReason-Bench**, a physics-grounded benchmark for evaluating generalist agentic planners on heterogeneous space planning problems. Our results show that while LLM agents demonstrate strong zero-shot adaptability, they remain limited in resource management and long-horizon spatial reasoning. **AstroReason-Bench** offers a realistic testbed for advancing agentic planning under strict physical constraints.

Limitations

This study establishes a baseline for agentic space planning, but several limitations remain.

Cost vs. Capability Trade-off Our evaluation focuses on efficient "Flash"-class models and a fixed per-episode budget (2-hour timeout with bounded CPU/memory). We did not benchmark larger, reasoning-intensive models (e.g., Claude Opus 4.5, Gemini 3 Pro) at the same scale due to prohibitive cost under hundreds of agent–environment interactions per mission. Future work will extend coverage to stronger models and report cost–performance trade-offs more systematically.

Operational Simplifications While our simulator enforces rigorous astrodynamics via SGP4, it abstracts away operational non-idealities such as component faults, thermal constraints, and stochastic effects (e.g., cloud cover for optical imaging). These choices isolate *planning* from *control* and enable controlled diagnosis, but they reduce realism relative to end-to-end operations.

Agent Scaffolding and Workflow Design We evaluate agents using a standard ReAct loop. More advanced scaffolding (e.g., explicit planning phases, verification, search, or learned controllers) may change the observed performance profile. Our results should therefore be interpreted as characterizing contemporary off-the-shelf agentic workflows rather than the upper bound of what is achievable with substantial agent engineering.

Statistical Coverage and Variance Each model is evaluated on a limited number of scenarios per benchmark. Given the stochasticity of LLM inference and tool-using behaviors, our reported averages may not fully capture variance across prompts, decoding settings, or random seeds. Increasing the number of episodes and reporting confidence intervals is an important direction for strengthening statistical conclusions.

Compute-Matched Comparisons We compare generalist agents against traditional baselines and, for SatNet, against published results from specialized optimizers. These comparisons are not compute-matched: specialized methods may leverage extensive offline optimization or training, whereas agents operate under a fixed online interaction budget. Our goal is not to claim optimality under equal compute, but to provide a diagnostic benchmark for adaptability and feasibility under realistic deployment constraints.

Scope Expansion Currently, AstroReason-Bench focuses on *operational* scheduling. A natural extension is *architectural* design, such as constellation design or deep-space trajectory planning, moving from resource management toward broader system engineering.

Ethics Statement

AstroReason-Bench is a benchmarking suite for evaluating LLM-based agentic planning in physics-constrained *Space Planning Problems (SPP)*. Our study is primarily diagnostic: it measures current agents' capabilities and failure modes under hard physical and operational constraints, rather than proposing deployment-ready autonomy for safety-critical missions.

Data, Licensing, and Privacy AstroReason-Bench is constructed from (i) publicly available orbital elements (Two-Line Elements, TLEs) and (ii) procedurally generated scenarios (targets, missions, and requests). The benchmark does not include personally identifiable information (PII) by design. Any auxiliary geographic target lists (e.g., city locations) are used only as generic coordinates; we do not associate them with individuals or sensitive attributes. We will release the benchmark code and datasets with clear documentation of upstream licenses for all external resources used (e.g., *TLE source and license*, *city database source and license*).

Human Participation No human subjects were recruited and no human annotations were collected. All evaluations are automated agent-environment interactions within a simulator. Consequently, this work does not involve IRB review or human-subject risks.

Model Use and Compliance We evaluate both open- and closed-source LLM systems via their official interfaces and in accordance with their respective terms of use. For locally executed open-weight models, we follow the corresponding licenses. We report aggregate benchmark metrics and qualitative traces necessary for scientific analysis, avoiding disclosure of proprietary model internals.

Safety, Dual Use, and Responsible Release Space mission planning can be considered dual-use. To mitigate misuse, AstroReason-Bench fo-

cuses on high-level scheduling and resource allocation abstractions rather than providing operational procedures for real systems. The simulator is not a drop-in controller for spacecraft or ground infrastructure, and the provided tools are limited to benchmark-relevant functions (state inspection, feasibility checking, and action staging) within a sandboxed environment. We will include a responsible use notice in the release, clarifying that the benchmark is intended for research on planning and verification, not for operational deployment without rigorous validation, oversight, and safety engineering.

AI-Assisted Tools We may use AI-assisted tools (e.g., code completion or language polishing) to improve engineering productivity and writing clarity. All benchmark implementations, experimental scripts, and reported results are manually reviewed by the authors. We also conduct spot checks to ensure that released artifacts do not contain secrets, PII, or harmful content.

Environmental Impact Benchmarking frontier models can incur non-trivial computational cost. We mitigate this by using fixed evaluation budgets (timeouts, CPU/memory caps) and reporting these settings to support reproducibility and facilitate fair cost-aware comparisons.

References

Marco Bagnardi, Pablo J González, and Andrew Hooper. 2016. High-resolution digital elevation model from tri-stereo pleiades-1 satellite imagery for lava flow volume estimates at fogo volcano. *Geophysical Research Letters*, 43(12):6267–6275.

Yuxuan Cai, Lu Chen, Qiaoling Chen, Yuyang Ding, Liwen Fan, Wenjie Fu, Yufei Gao, Honglin Guo, Pinxue Guo, Zhenhua Han, and 1 others. 2025. Nex-n1: Agentic models trained via a unified ecosystem for large-scale environment construction. *arXiv preprint arXiv:2512.04987*.

Xiaoli Cao, Yitao Li, Xingzhong Xiong, and Jun Wang. 2022. Dynamic routings in satellite networks: An overview. *Sensors*, 22(12):4552.

Thomas Claudet, Ryan Alimo, Edwin Goh, Mark D Johnston, Ramtin Madani, and Brian Wilson. 2022. δ-milp: Deep space network scheduling via mixed-integer linear programming. *IEEE Access*, 10:41330–41340.

JT Dolloff and HJ Theiss. 2012. Temporal correlation of metadata errors for commercial satellite images: Representation and effects on stereo extraction accuracy. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 39:215–223.

Edwin Goh, Hamsa Shwetha Venkataram, Bharathan Balaji, Brian D Wilson, and Mark D Johnston. 2021. Satnet: A benchmark for satellite scheduling optimization. In *AAAI-22 Workshop on Machine Learning for Operations Research (ML4OR)*.

Alexandre Guillaume, Seugnwon Lee, Yeou-Fang Wang, Hua Zheng, Robert Hovden, Savio Chau, Yu-Wen Tung, and Richard J Terrile. 2007. Deep space network scheduling using evolutionary computational methods. In *2007 IEEE Aerospace Conference*, pages 1–6. IEEE.

Adam Herrmann and Hanspeter Schaub. 2023. Reinforcement learning for the agile earth-observing satellite scheduling problem. *IEEE Transactions on Aerospace and Electronic Systems*, 59(5):5235–5247.

Felix R Hoots and Ronald L Roehrich. 1980. Models for propagation of norad element sets.

Xiaoxuan Hu, Waiming Zhu, Huawei Ma, Bo An, Yanling Zhi, and Yi Wu. 2021. Orientational variable-length strip covering problem: A branch-and-price-based algorithm. *European Journal of Operational Research*, 289(1):254–269.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. Swe-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*.

Mark D Johnston and Bradley J Clement. 2006. Automating deep space network scheduling and conflict resolution. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 1483–1489.

Mark D Johnston, Daniel Tran, Belinda Arroyo, and Chris Page. 2009. Request-driven scheduling for nasa's deep space network.

Patrick W Kenneally, Scott Piggott, and Hanspeter Schaub. 2020. Basilisk: A flexible, scalable and modular astrodynamics simulation framework. *Journal of aerospace information systems*, 17(9):496–507.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Soung Sub Lee, Jong Pil Kim, Eungnoh You, Jae-Hyuk Youn, and Ho-Hyun Shin. 2024. Satellite constellation method to achieve desired revisit performance for multiple targets. *Journal of Applied Remote Sensing*, 18(2):024509–024509.

Michel Lemaître, Gérard Verfaillie, Frank Jouhaud, Jean-Michel Lachiver, and Nicolas Bataille. 2002. Selecting and scheduling observations of agile satellites. *Aerospace Science and Technology*, 6:367–381.

Tianzuo Li and Guangyuan Wang. 2025. A scheduling method for real-time multi-fold regional coverage based on meo constellations. *Advances in Space Research*.

Wei Li, Xin Zhang, Zhongxin Guo, Shaoguang Mao, Wen Luo, Guangyue Peng, Yangyu Huang, Houfeng Wang, and Scarlett Li. 2025. Fea-bench: A benchmark for evaluating repository-level code generation for feature implementation. *arXiv preprint arXiv:2503.06680*.

XM Li. 2017. Two-archive2 algorithm for large-scale polygon targets observation scheduling problem. In *2017 2nd International Conference on Information Technology and Management Engineering*, pages 1–6.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Pete Florence, Andy Zeng, and 1 others. Code as policies: Language model programs for embodied control. In *Workshop on Language and Robotics at CoRL 2022*.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.

Yifeng Lyu, Han Hu, Rongfei Fan, Zhi Liu, Jianping An, and Shiwen Mao. 2024. Dynamic routing for integrated satellite-terrestrial networks: A constrained multi-agent reinforcement learning approach. *IEEE Journal on Selected Areas in Communications*, 42(5):1204–1218.

Martínez Contreras Johana Milena, Pantoja Benavides Germán Fernando, Astrid Xiomara Rodríguez, John Willmer Escobar, and David Álvarez-Martínez. 2025. Exact and heuristic algorithms for convex polygon decomposition. *Mathematics*, 13(24):4038.

Davide Paglieri, Bartłomiej Cupiał, Samuel Coward, Ulyana Piterbarg, Maciej Wolczyk, Akbir Khan, Eduardo Pignatelli, Łukasz Kuciński, Lerrel Pinto, Rob Fergus, and 1 others. Balrog: Benchmarking agentic llm and vlm reasoning on games. In *The Thirteenth International Conference on Learning Representations*.

David Vallado, Paul Crawford, and Richard Hujsak. Revisiting spacetrack report# 3. In *AIAA/AAS astrodynamics specialist conference and exhibit*, page 6753.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2023a. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. *Advances in Neural Information Processing Systems*, 36:38975–38987.

Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. 2023b. On the planning abilities of large language models-a critical investigation. *Advances in Neural Information Processing Systems*, 36:75993–76005.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*.

Hui Wei, Zihao Zhang, Shenghua He, Tian Xia, Shijia Pan, and Fei Liu. 2025. Plangenllms: A modern survey of llm planning capabilities. *arXiv preprint arXiv:2502.11221*.

Ke Wu, Yasser Bigdeli, Seyed Ali Keivaan, Jie Deng, and Pascal Burasa. 2025. Integrated sensing and communication (isac) transceiver: Hardware architectures, enabling technologies, and emerging trends. *IEEE Journal of Selected Topics in Electromagnetics, Antennas and Propagation*.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. 2024. Travelplanner: A benchmark for real-world planning with language agents. In *International Conference on Machine Learning*, pages 54590–54613. PMLR.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. tau-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

LU Zezhong, Xin Shen, LI Deren, Dilong Li, Yaxin Chen, Di Wang, and Shuai Shen. 2023. Multiple super-agile satellite collaborative mission planning for area target imaging. *International Journal of Applied Earth Observation and Geoinformation*, 117:103211.

Zizheng Zhan, Ken Deng, Jinghui Wang, Xiaojiang Zhang, Huaixi Tang, Minglei Zhang, Zhiyi Lai, Haoyang Huang, Wen Xiang, Kun Wu, and 1 others. 2025. Kat-coder technical report. *arXiv preprint arXiv:2510.18779*.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, and 1 others. Webarena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations*.

A Appendix

A.1 Baseline Limitations

Baseline Limitations These baselines serve as reference implementations rather than optimized solvers. Key limitations include: (1) hyperparameters and heuristics are not carefully tuned for individual benchmarks; (2) the implementation is not optimized for high-throughput computation; and (3) each baseline run is limited to ∼20 minutes. Given additional computation, baseline performance would likely improve; for reference, MILP solutions in prior work required ∼20 hours of optimization (Claudet et al., 2022).

A.2 Additional Case Studies

We present two additional targeted case studies that probe specific cognitive capabilities required for space planning. Each study isolates a distinct failure mode, applies a minimal intervention, and measures the resulting behavioral change.

A.2.1 The Exploration-Exploitation Gap

Phenomenon In Regional Coverage, agents consistently achieved near-zero coverage despite the benchmark being theoretically solvable. Analysis revealed a common pattern: agents registered observation strips almost *immediately* after reading the mission brief, without first querying satellite ground tracks to understand orbital geometry.

Example In a representative Claude Sonnet 4.5 run in regional coverage case 1, where the agent is required to plan observations for three polygons (Amazon Basin, Gulf of Mexico, Bay of Bengal) with 15 satellites in SKYSAT[6] constellation, the agent's first action after querying satellites and stations was to register 5 strips within Bay of Bengal:

These random strips are highly inefficient and do not align with satellites' ground tracks, leading to limited access windows.

Intervention We re-ran the first case in Regional Coverage using Claude Sonnet 4.5 with Plan Mode manually enabled and an additional hint "Analyze available tools and reason about polygon decomposition strategy."

Outcome The agent produced a detailed planning document that correctly reasoned about orbital dynamics:

[6] https://earth.esa.int/eogateway/missions/skysat

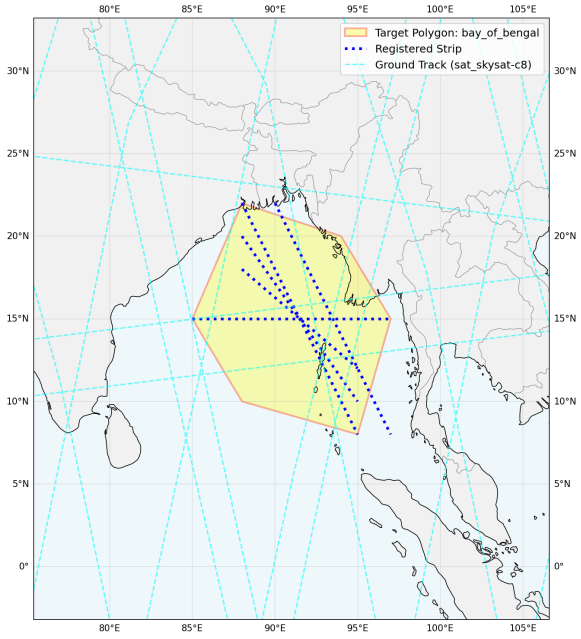

Figure 4: **Naive Decomposition.** The agent's initial coverage features 5 strips: they are inefficient, overlapping, and intersect with each other - basically random lines.

> *"Near-polar orbits (97–98° inclination) produce ground tracks that are predominantly N-S oriented, maximizing strip coverage efficiency. [...] Strip spacing = 5.0 km (12% overlap buffer for edge effects)."*

This led to N-S oriented strips aligned with satellite velocity vectors, which is a correct decomposition strategy. The final plan achieved **8% coverage**, a modest improvement over the baseline run (0%). However, the agent still did not query actual ground tracks via `get_ground_track()`, instead relying on general orbital knowledge. The remaining gap to optimal performance stems from (1) imprecise strip placement without ground track data, and (2) storage exhaustion from aggressive observation scheduling.

Implication Structured reasoning phases can unlock latent domain knowledge, but agents exhibit a persistent action bias, preferring to reason from memory rather than actively exploring the environment. Access to tools alone is insufficient; agents must be prompted to use exploratory tools before committing to strategies.

A.2.2 RAG-Enhanced Planning

Hypothesis Providing agents with domain-specific academic literature may improve strategic

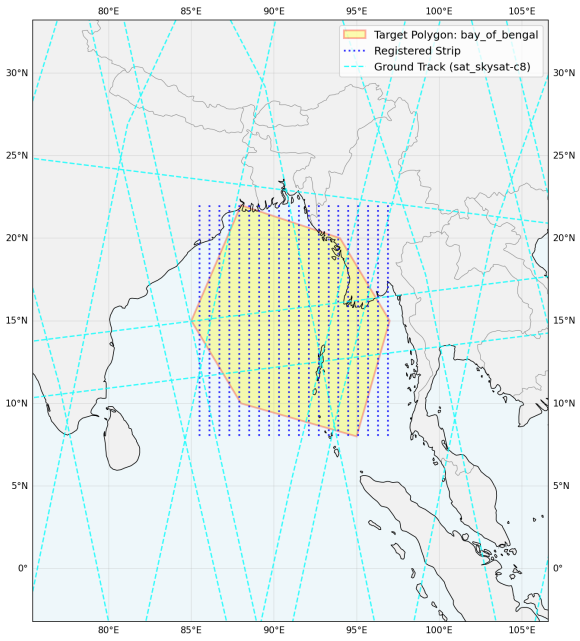

Figure 5: **Strategic Decomposition.** The planned coverage features 20 vertical strips, which align with the near N-S pattern of satellite ground tracks.

planning by exposing effective algorithm patterns.

Experiment We re-ran the case of SatNet Week 40 (`W40_2018`), the most difficult case characterized by extreme oversubscription (Claudet et al., 2022), using Claude Sonnet 4.5. We injected markdown versions of relevant academic papers into the workspace and appended the prompt with "Note: The `related_works/` folder contains research papers that may provide useful insights and approaches." We compared two conditions: **default mode** (autonomous) and **plan mode** (needs manual triggering and plan approval).

Outcome In **default mode**, the agent exhibited a strong action bias, skimming only fragments of one to two papers before acting. This often degraded performance: reading about the problem's difficulty led to early resignation while reading about the high baseline scores led to brute-force retries without strategic improvement. The "related_works" effectively became noise. However, in **plan mode**, the agent engaged deeply with the literature, synthesizing a hybrid strategy from multiple sources. It correctly identified that "Systematic backtracking works for small regions" but "MILP with randomization" is needed for full schedules. It proposed and implemented a nuanced algorithm:

> *"1. Use MILP randomization for initial schedule (fairness + quality); 2. Apply backtracking to resolve conflicts [...] 3. Use greedy extension for unused antenna time."*

This RAG+Plan approach yielded significantly better scores ($U_{rms} \approx 0.50$) than default runs.

Implication Access to knowledge is insufficient; agents need structured workflows instead of raw ReAct loop to consume it.