SwarmSAT Agent: A Tool-Grounded Autonomous Multi-Agent LLM Framework for Rapid Satellite Constellation Retasking of Pop-Up Events

Anonymous CVPR submission

Paper ID *****

Abstract

Coordinating a satellite constellation for time-critical pop-up events (e.g., wildfires) requires fast, resource-aware tasking under partial visibility, limited onboard energy, and operator-imposed rules. We present SwarmSAT, a physically grounded multi-agent constellation tasking framework that treats pop-up response as a lightweight sequential handoff auction. A Cluster Lead agent detects a pop-up, identifies eligible satellites via orbital visibility windows derived from real TLE propagation, and orchestrates turn-based bid submission from satellite agents. Each satellite computes a composite bid that trades off predicted peak imaging quality against current battery state, enabling resource-aware assignment and explicit redundancy suppression. We evaluate SwarmSAT in a discrete-time orbital simulation with battery recovery/consumption dynamics, routine background targets, and injected pop-up disasters, and compare against a greedy baseline scheduler. In a representative Australia bushfire scenario, SwarmSAT eliminates redundant pop-up imaging (0% redundant vs. 98.6% for greedy) and reduces time-to-first observation from 720 s to 60 s. Across four geographically distinct pop-up scenarios, SwarmSAT improves average pop-up utility by ∼36% while reducing time-to-first observation by ∼99%, and it maintains near-zero redundancy by selecting a single winner to execute the task. Beyond performance, SwarmSAT enables controllable operations where it satisfies 4/4 natural-language operator constraints (battery floor, quality floor, satellite exclusion, and no-go zone avoidance), whereas the greedy baseline violates 3/4. We further report a multi-pop-up stress test illustrating degradation under simultaneous disasters, motivating coordination mechanisms for scalable real-world event response.

1. Introduction

Time-critical pop-up events (e.g., wildfires, volcanic eruptions, floods, fast-evolving storms) increasingly drive demand for rapid, high-quality satellite observations, often under hard physical constraints such as time-varying access windows and viewing geometry, and operational constraints such as limited energy budgets and operator policies. As constellations scale, the retasking problem becomes both more urgent and more complex as schedules must be revised online, decisions must be made with incomplete information, and coordination is required to avoid wasted effort when multiple platforms independently converge on the same urgent target. These pressures have motivated decades of work on automated planning, scheduling, and autonomy for space operations, and they are now re-emerging in modern large-constellation settings where responsiveness and coordination are first-class requirements.

Historically, satellite tasking began as predominantly ground-driven and heavily scripted, with operators constructing imaging campaigns by hand and relying on local heuristics to respect timing and resource constraints. As mission complexity increased, automated planning began to enter operations, exemplified by systems that generated large volumes of feasible imaging "data takes" under coupled constraints, such as the RADARSAT mission planner used for complex mapping campaigns [17]. With growing mission demand and oversubscription, the field shifted toward algorithmic scheduling methods designed to handle many competing requests, heterogeneous constraints, and limited onboard/ground resources. Constraint-based formulations and heuristic search became central in early fleet-scheduling work for Earth-observing missions [6], while alternative optimization paradigms such as evolutionary algorithms were explored to cope with the combinatorial structure of large scheduling instances [8]. Contemporary surveys reaffirm that satellite scheduling remains an active area because constellation-scale operations amplify both the problem size and the diversity of constraints across imaging, communication, and multi-mission settings [5].

A second major driver has been the rise of unpredictable, high-value targets-of-opportunity (ToOs), which require inserting new tasks and repairing plans on short notice. This setting motivated continuous planning and replanning archi-

tectures, notably CASPER (Continuous Activity Scheduling Planning Execution and Replanning), which emphasized persistent plan repair as new information arrives and execution deviates from the plan [12]. The push toward event-driven responsiveness culminated in demonstrations of onboard autonomy for science, such as the EO-1 Autonomous Science Agent, which integrated onboard detection with replanning to revise observation plans in response to newly discovered phenomena [3]. In parallel, mixed-initiative systems established an operational doctrine that autonomy must remain operator-steerable by incorporating human intent and constraints while automation handles feasibility and optimization; MAPGEN is a canonical example of this principle in planetary operations [1]. Together, these developments show that fast replanning is feasible and valuable, but also that practical deployments must handle real constraints, uncertainty, and human oversight.

Beyond centralized planning, distributed and market-based coordination has long provided a principled abstraction for allocating tasks among multiple agents under resource constraints. Auction mechanisms in particular offer scalable and interpretable allocation procedures when foundational results such as the auction algorithm show how bidding can solve assignment problems efficiently in distributed settings [2]. In multi-agent and multi-robot coordination, formal analyses and surveys describe how task allocation decomposes into structured variants and how market-based approaches trade off optimality, robustness, and scalability [4, 7]. Classic economic results (e.g., Vickrey auctions) underpin truthful mechanisms and allocation under uncertainty [20], while combinatorial auctions formalize richer winner-determination problems for bundled tasks at increased computational cost [14]. These ideas are directly relevant to constellation response, where decisions must often be made quickly, using simple and auditable signals, and where distributed coordination can reduce redundant actions and improve global utility.

Most recently, large language models (LLMs) have enabled a different style of decision-making in which a controller can combine natural-language reasoning with explicit interactions through tools and structured state. Research on tool-using LLMs has shown that interleaving reasoning with environment actions can improve grounding and reduce unfaithful free-form claims [21], and that models can learn when to invoke external tools rather than relying only on parametric memory [16]. Modular neuro-symbolic perspectives argue for composing LLMs with external solvers and knowledge sources to improve reliability and controllability [9]. In parallel, agent orchestration frameworks have emerged to support long-running, stateful workflows with features such as durable execution and human-in-the-loop control [13, 15]. For time-critical constellation operations, these developments suggest a promising direction in which tool-grounded agent controllers can reason over constraints, call physics-based simulators, and produce auditable decisions under operator rules.

In this work, as can be seen in Figure 1, we present SwarmSAT, a LangGraph-orchestrated LLM multi-agent controller for rapid pop-up response in Earth-observation constellations. SwarmSAT uses a sequential handoff auction where a Cluster Lead agent forecasts physically grounded orbital access windows from TLEs using SGP4-compatible propagation [10, 11, 18, 19], solicits bids from satellite agents, and deterministically assigns a single winner per pop-up to minimize latency and prevent wasteful multi-satellite duplication. Each satellite produces an auditable bid from predicted best viewing geometry (minimum off-nadir angle) and local energy state, with battery evolving under a recharge-and-spend model; the protocol is hardened with idempotent auction/bid semantics to reduce brittleness under retries common in streaming agent execution.

2. Proposed Approach

We propose SwarmSAT, a physically grounded, tool-driven multi-agent controller for rapid constellation retasking in response to time-critical pop-up events. The core idea is to treat pop-up response as an online single-target distributed assignment problem executed at detection time. Rather than allowing each satellite to independently pursue the same high-priority event, SwarmSAT coordinates the constellation through a lightweight sequential handoff auction in which a single Cluster Lead agent queries physics-based orbital access, solicits bids from satellite agents, and deterministically assigns exactly one winner to execute the observation. All decisions are grounded through explicit tool calls to a shared simulation environment that exposes visibility, geometry, energy state, bids, and logs; consequently, the method does not rely on unverified natural-language claims to compute feasibility or outcomes.

2.1. Orbital access model and viewing geometry

SwarmSAT grounds visibility and imaging geometry in two-line element (TLE) ephemerides propagated with an SGP4-compatible pipeline. According to Figure 2, at a simulation time t, each satellite s has a sub-satellite position (latitude, longitude, altitude) obtained from propagation, and each ground target τ is represented by geodetic latitude/longitude. Visibility is determined by the satellite-to-target topocentric elevation angle $\mathrm{el}_{s,\tau}(t)$; a target is visible if $\mathrm{el}_{s,\tau}(t) \geq \mathrm{el}_{\min}$ for a configured minimum elevation threshold. For each visible target, we compute an off-nadir (view) angle $\theta_{s,\tau}(t)$ that serves as a proxy for achievable image quality, with smaller θ indicating more nadir-like viewing and higher quality. In addition to instantaneous geometry, the controller requires near-term access forecast-

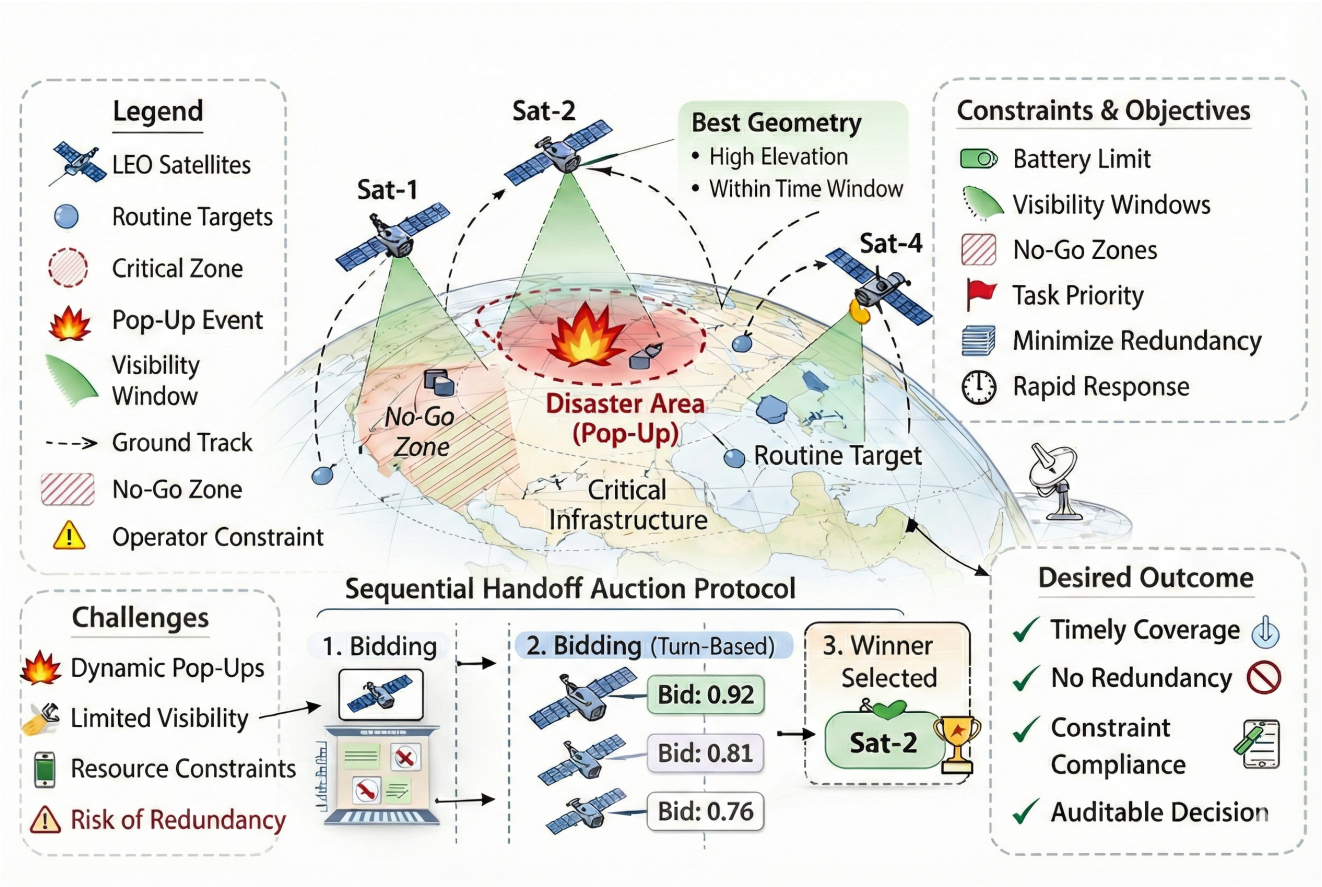

Figure 1. Overview of SwarmSAT where a Cluster Lead queries physics-based access, orchestrates a sequential handoff auction, and assigns a single winning satellite to observe the pop-up.

ing. For a lookahead horizon H minutes, we compute discrete-time visibility windows $\mathcal{W}_{s,\tau} = \{[t^{(k)}_{\text{start}}, t^{(k)}_{\text{end}}]\}_k$ by scanning forward in time at the simulation step resolution and grouping contiguous visible intervals; within each window, we track the best achievable geometry $\theta^{\star}_{s,\tau} = \min_{t\in[t_{\text{start}},t_{\text{end}}]} \theta_{s,\tau}(t)$ and the corresponding maximum elevation. This window-level statistic is the basis for bidding and enables the Cluster Lead to identify the eligible set of satellites that can service the pop-up within the lookahead.

2.2. Pop-up injection and energy dynamics

We model constellation operations in a discrete-time simulation environment with step size Δt minutes. The environment maintains a catalog of routine targets and supports the injection of pop-up events. A pop-up target becomes active at a specified time step and is exposed to the agents through the environment state returned by tool calls. In our implementation, pop-ups are injected and become active in the subsequent simulation step, matching the operational reality that detection precedes tasking by at least one control cycle. Each satellite carries an energy state represented by a battery percentage $B_s(t) \in [0, 100]$. Battery evolves under a recharge-and-spend model that captures (i) continuous recovery at rate ρ percent per minute and (ii) a fixed observation cost κ percent per image. Let $\mathbb{I}_s(t) \in \{0, 1\}$ denote whether satellite s performs an observation at time step t. Then the battery update is

$$B_s(t+\Delta t) \;=\; \text{clip}\Big(B_s(t) + \rho\,\Delta t - \kappa\,\mathbb{I}_s(t),\; B_{\min},\; B_{\max}\Big), \tag{1}$$

where $\text{clip}(x, B_{\min}, B_{\max}) = \min\{B_{\max}, \max\{B_{\min}, x\}\}$ and $B_{\min}, B_{\max}$ are configured limits. This simple dynamic is sufficient to induce the key operational trade-off where satellites with superior geometry may be temporarily ineligible due to low energy, and coordinated tasking should avoid exhausting batteries through redundant imaging.

2.3. Sequential handoff auction and coordination

As can be observed in Figure 3, SwarmSAT instantiates one agent per satellite and a Cluster Lead agent within a LangGraph state machine. The Cluster Lead orchestrates

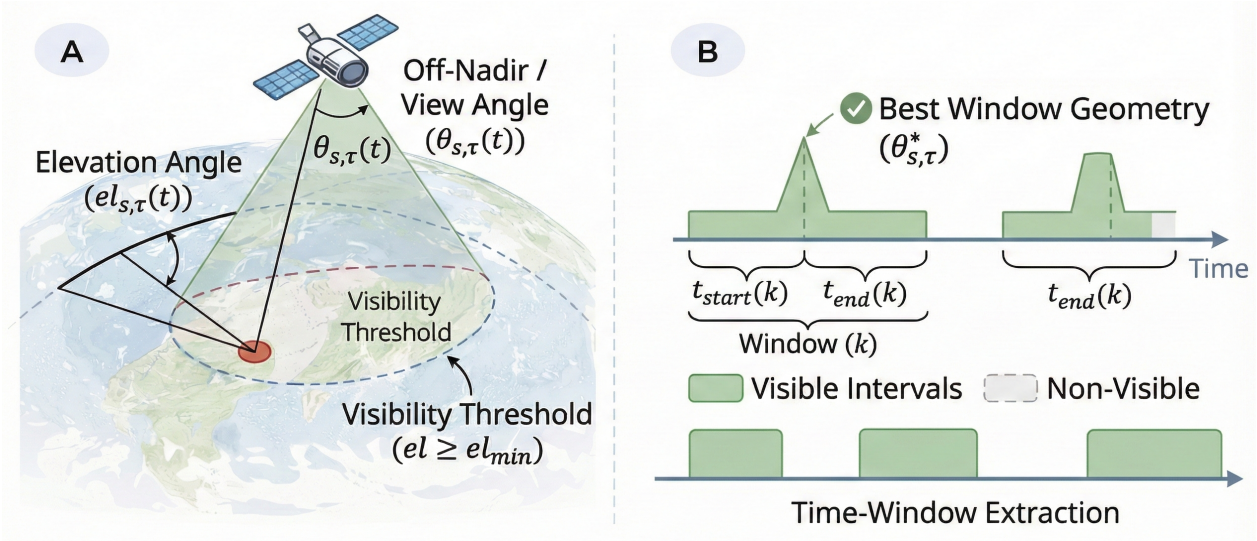

Figure 2. Geometry and visibility-window computation when (A) instantaneous elevation $el_{s,\tau}(t)$ and off-nadir angle $\theta_{s,\tau}(t)$ define visibility and (B) contiguous visible intervals form access windows, and the best window-level angle $\theta^*_{s,\tau}$ is used for bidding.

response when a pop-up is detected. At detection time, the Cluster Lead (i) queries the environment for newly active pop-ups, (ii) requests access forecasts to determine which satellites have a feasible window within a specified lookahead horizon, and (iii) runs a sequential handoff auction by contacting each eligible satellite agent in turn. Each satellite agent queries its own local state from the environment (including battery and predicted best viewing geometry for the pop-up), then submits a bid via a dedicated tool call. The Cluster Lead collects all bids, selects the winner by a deterministic rule (highest bid), and commits the assignment through the environment schedule.

A key design principle is that all multi-agent interaction is mediated by the environment's tool API, which acts as a shared, auditable source of truth. Specifically, bid submission, auction state queries, and winner assignment are written to environment-side state rather than relying solely on transient conversational memory. To improve robustness under tool retries or duplicated calls, the environment enforces idempotent auction semantics for which starting an auction for a target that is already active is treated as a no-op, and duplicate bids for the same (satellite, target) pair are ignored. These guards preserve consistency without requiring perfect agent behavior.

2.4. Bid construction from geometry and energy

To maintain interpretability and auditability, we use a transparent bid function that explicitly trades off imaging geometry and available energy. Let $\theta^\star_{s,\tau}$ denote the best (minimum) off-nadir angle predicted for satellite s to observe pop-up target τ within the lookahead window, and let B_s be the satellite's battery percentage at bid time. We map geometry to a normalized quality proxy as:

$$q_{s,\tau} \;=\; 1 - \frac{\theta^\star_{s,\tau}}{90^\circ}, \tag{2}$$

and define the composite bid as:

$$b_{s,\tau} \;=\; q_{s,\tau} \cdot \frac{B_s}{100}. \tag{3}$$

This form has three desirable properties for online tasking as it is fast to compute, bounded in $[0, 1]$, and directly interpretable by operators as a product of "how good the image can be" and "how much energy the satellite can spare." In our experiments, this bid captures qualitatively correct behaviors observed in real operations where a slightly inferior geometric opportunity may be preferred when it preserves energy or when the best-geometry satellite is energy-constrained, and the auction naturally suppresses redundant observations by selecting a single winner.

2.5. Observation and redundancy semantics

After winner assignment, the environment records the executed observation in a structured log entry containing (s, τ), timestamp, and quality flag. Duplicate detection is global where an observation of target τ is marked duplicate if τ has been observed previously by any satellite. This explicit semantics enables the direct measurement of wasted effort in pop-up response. In addition, all auction events (turn-taking, bids, winner selection) are captured in a negotiation transcript through a dedicated logger. This transcript supports both qualitative analysis of the decision process and quantitative verification that the implemented protocol matches the intended design.

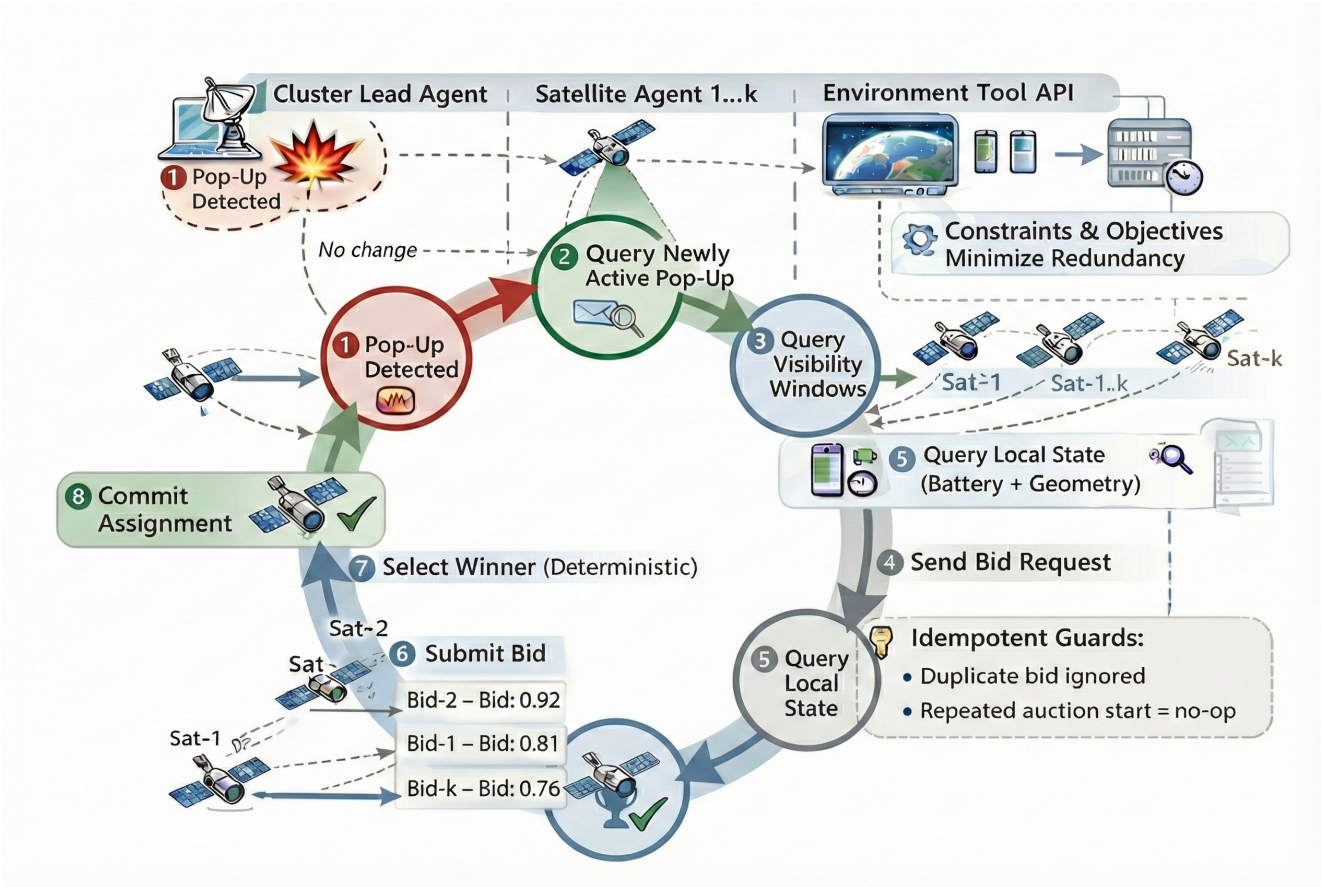

Figure 3. The overview schematic of sequential handoff auction protocol.

2.6. Operator constraints as executable policies

A central requirement for operational autonomy is controllability under human intent. SwarmSAT supports natural-language operator constraints that restrict feasible actions at decision time (e.g., minimum battery floor, maximum allowable view angle, exclusion of specific satellites, and avoidance of no-go zones). In the proposed approach, constraints are injected into the agent context and enforced through tool-grounded checks for which satellites can abstain from bidding when constraints would be violated, and the Cluster Lead can reject infeasible bids or assignments. This design yields two benefits: constraints remain legible to operators, and compliance is verifiable from environment logs. Empirically, this mechanism is responsible for the sharp compliance gap observed between SwarmSAT and greedy baselines in the constraint suite.

2.7. Baselines and experimental protocol

We compare SwarmSAT against a greedy baseline scheduler that, at each time step, independently selects for each satellite the highest-priority visible target. This baseline is intentionally simple and widely used as a proxy for local decision-making under time pressure. However, because it lacks explicit coordination, it tends to drive multiple satellites toward the same urgent pop-up and to produce extreme redundancy when the pop-up dominates local priorities. To ensure fair comparison in the pop-up setting, we evaluate both methods on pop-up-only metrics: we compute performance with respect to the injected pop-up target(s), measured from simulator logs, rather than conflating routine-target imaging with emergency response.

2.8. Metrics and simulator logs

SwarmSAT is evaluated using three primary metrics computed purely from environment logs, enabling reproducibility and eliminating reliance on self-reported agent outputs. Let $\mathcal{O}$ be the set of all observations, and let $\mathcal{T}$ be the set of targets. For each target τ, define the best achieved quality $Q(\tau) = \max_{(s,\tau)\in\mathcal{O}} q_{s,\tau}$, with $Q(\tau) = 0$ if τ is never observed. Given a target priority weight $w(\tau) \in [0, 1]$, we define the global utility as:

$$U = \sum_{\tau\in\mathcal{T}} w(\tau)\, Q(\tau). \tag{4}$$

Redundancy ratio measures the fraction of observations that repeat an already-observed target,

$$R = \frac{1}{|\mathcal{O}|} \sum_{o \in \mathcal{O}} \mathbb{I}\{o \text{ is duplicate}\}, \tag{5}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function. Adaptability quantifies responsiveness to a pop-up: if t_{detect} is the pop-up activation time and t_{first} is the timestamp of the first successful observation of that pop-up, then

$$A = t_{\text{first}} - t_{\text{detect}}. \tag{6}$$

For multi-pop-up stress tests, we report coverage rate as the fraction of pop-ups observed at least once and average adaptability across pop-ups with successful observations. These metrics directly correspond to operational objectives that include maximization of useful information, minimizing wasted effort, and minimizing time-to-first response.

3. Data and Scenario Construction

Our study is driven by a simulator-centric dataset rather than a pre-labeled benchmark corpus. Specifically, the data used in this work consist of (i) orbital state inputs for the satellite constellation, (ii) a geospatial target catalog with static and dynamic (pop-up) events, and (iii) execution traces generated by the baseline and multi-agent controllers during simulation. This design is appropriate for time-critical constellation tasking because the central problem is online coordination under physical and operational constraints, not supervised prediction from fixed labels.

At the constellation layer, each satellite is represented by a Two-Line Element (TLE) record, and the simulator propagates orbits using an SGP4-compatible pipeline through Skyfield-based routines. Let

$$\mathcal{S} = \{s_1, \ldots, s_{N_s}\}, \qquad N_s = 10,$$

denote the active constellation used in the main experiments. For each satellite s_i, the orbital input is a tuple as:

$$e_i = (\ell_{i1}, \ell_{i2}),$$

where ℓ_{i1} and ℓ_{i2} are the two TLE lines. The implementation supports loading TLEs from an external source (CelesTrak/Planet feed) and also includes an embedded fallback TLE set to ensure reproducibility and offline execution. The simulator assigns logical IDs $\texttt{SAT_0}, \ldots, \texttt{SAT_9}$ to these propagated satellites. This separation between physical ephemerides and logical IDs is important because it allows the agent system to reason over symbolic satellite identities while all geometry remains grounded in orbital propagation. At the target layer, we use a mixed geospatial catalog,

$$\mathcal{T}(t) = \mathcal{T}_{\text{static}} \cup \mathcal{T}_{\text{popup}}(t),$$

where $\mathcal{T}_{\text{static}}$ contains persistent targets and $\mathcal{T}_{\text{popup}}(t)$ contains urgent events injected during runtime. Each target is encoded as:

$$\tau_j = \left(\text{id}_j, \text{name}_j, \phi_j, \lambda_j, p_j, \kappa_j, t_j^{\text{appear}}\right),$$

with latitude ϕ_j, longitude λ_j, priority $p_j \in [0, 1]$, type $\kappa_j \in \{\texttt{routine}, \texttt{critical}, \texttt{popup}\}$, and an appearance timestamp t_j^{appear} (used for pop-up events). In the provided target CSV/code generation pipeline, the static catalog combines globally distributed routine city targets with a set of fixed critical disaster-prone regions. The routine targets receive seeded random priorities in the range $[0.1, 0.5]$, while critical targets are assigned higher fixed priorities (e.g., 0.8–1.0). In the current generated CSV used for this project, the catalog contains 106 static targets (101 routine and 5 critical), which is consistent with the shipped target file, even though one code docstring describes the intended set as 105 targets. This distinction is worth documenting for reproducibility.

The dynamic pop-up data are generated procedurally from a predefined set of high-value emergency candidates (e.g., wildfire/cyclone locations), each injected with a fixed high priority ($p = 0.95$) and a runtime appearance timestamp. If $t_k = t_0 + k\Delta t$ denotes the simulation clock, then pop-ups are introduced by augmenting the active target set at a designated step k_{popup}:

$$\mathcal{T}_{\text{popup}}(t_k) = \begin{cases} \varnothing, & k < k_{\text{popup}}, \\ \{\tau_1^{\text{popup}}, \ldots, \tau_m^{\text{popup}}\}, & k \geq k_{\text{popup}}. \end{cases}$$

This runtime insertion mechanism is central to the problem formulation because it forces the controller to reallocate sensing resources online without restarting the mission plan. The simulator itself defines the temporal sampling used to convert orbital and geospatial inputs into visibility data. We use a fixed-step horizon with simulation start time t_0 (UTC), step size $\Delta t = 1$ minute, and total duration $T = 6$ hours. For each pair (s_i, τ_j), the orbital engine evaluates visibility over sampled times and constructs contiguous visibility windows whenever the target elevation exceeds the minimum threshold (10° in the current configuration). Denoting satellite elevation above the local horizon by $\alpha_{ij}(t)$, visibility is declared when

$$\alpha_{ij}(t) \geq \alpha_{\min}, \qquad \alpha_{\min} = 10^\circ.$$

The simulator then stores for each window its start/end times, maximum elevation, and the corresponding minimum off-nadir angle. These window records form the physically grounded data consumed by the bidding and assignment logic. In addition to orbital and target inputs, the simulation generates internal state data needed for realistic coordination. Each satellite carries a battery state $B_i(t)$ initialized from a seeded random distribution in $[30, 70]\%$, then

updated through recharge and observation cost dynamics. At each simulation increment,

$$B_i(t+\Delta t) = \min(B_{\max},\, B_i(t) + r\Delta t) - c\, n_i(t, t+\Delta t),$$

where $r = 1\%$/min is the recovery rate, $c = 5\%$ is the per-observation battery cost, and $n_i(t, t+\Delta t)$ is the number of photographs executed by satellite s_i during that interval. This battery trajectory is not merely a simulator variable; it is a core input to the bid computation and therefore part of the effective decision-making dataset. The data used for evaluation are generated as structured logs from controller execution. For each run, we record observations,

$$o = (\text{satellite_id}, \text{target_id}, q, t, \text{is_duplicate}),$$

bid submissions (including bid value, battery percentage, and view angle), assignment events (winner plus bid set), and full negotiation transcripts (agent turns, tool calls, and timestamps). These logs are then post-processed into the reported metrics (utility, redundancy, adaptability, coverage, and constraint compliance). Importantly, this means that our reported results are traceable to concrete execution traces rather than aggregate counters only, which improves auditability of the multi-agent decision process. To probe generalization and failure modes, we also construct several scenario datasets from the same simulator primitives. The robustness sweep varies the pop-up geography across four predefined emergency locations while keeping the constellation and timing fixed. The scalability sweep varies constellation size ($N_s \in \{3, 5, 7, 10\}$) and re-runs the same pop-up insertion protocol. The stress test constructs simultaneous-pop-up datasets with $m \in \{1, 2, 3, 5\}$ concurrent emergencies injected at the same time step. Finally, the constraint-following experiments augment the same physical environment with natural-language operator rules (e.g., battery floor, quality floor, satellite exclusion, and restricted-zone avoidance), and compliance is evaluated post hoc from the generated bid/assignment/observation logs. Taken together, these scenario datasets allow us to evaluate not only nominal performance but also robustness, scaling behavior, and controllability under operator directives. More details about the dataset used in this study can be found in Appendix (A).

4. Results and Discussion

We evaluated SwarmSAT on a 6-hour simulation with 10 LEO satellites, 1-minute temporal resolution, 100 routine city targets, 4 critical zones, and one injected pop-up event in simulation step 30. The results are reported from structured simulator logs, bid traces, and negotiation transcripts, and are summarized in Table 1 as it displays the quantitative analysis of bids submitted by all eligible satellites during the Australia Bushfire pop-up event. Each bid is computed from the satellite's best achievable viewing geometry,

Table 1. Results summary of the SwarmSAT simulation and evaluation constraints.

Satellite	Nadir (°)	Battery (%)	Bid Value	Cost (%)	Status
SAT_5	27.6	100.0	0.6933	5.0	WINNER
SAT_7	26.7	92.4	0.6510	5.0	
SAT_0	28.4	92.0	0.6295	5.0	
SAT_2	31.7	95.3	0.6171	5.0	
SAT_3	27.8	88.9	0.6142	5.0	
SAT_6	31.6	91.4	0.5930	5.0	
SAT_9	25.0	79.0	0.5706	5.0	
SAT_1	30.4	78.6	0.5206	5.0	
SAT_8	23.9	66.1	0.4851	5.0	
SAT_4	26.1	64.8	0.4603	5.0	

represented by the nadir angle, and its current battery level according to the bid function defined in Section 2. Satellites with smaller nadir angles and higher battery reserves obtain higher bid values, reflecting both observation quality and resource availability. In this instance, SAT_5 achieves the highest bid value of 0.6933 due to its full battery state and favorable viewing geometry, and is therefore selected as the deterministic winner. All other satellites remain eligible but are not assigned the task. This table illustrates how the auction mechanism transparently prioritizes satellites based on physically meaningful state variables while ensuring a single, non-redundant assignment.

In term of robustness study, Table 2 reports the performance of the proposed SwarmSAT controller compared with a greedy baseline across multiple pop-up disaster scenarios. The SwarmSAT approach achieves substantially higher global utility (+36.3%) while completely eliminating redundant pop-up observations, reducing the redundancy ratio from 0.9803 to 0.0000. Adaptability is dramatically improved, with response latency reduced from an average of 5025 s to 60 s, indicating significantly faster task allocation following pop-up activation. Additionally, the total number of pop-up observations decreases from 54.25 to 1.0, demonstrating that SwarmSAT assigns exactly one optimal satellite rather than allowing multiple redundant observations. The per-scenario breakdown shows consistent performance across geographically diverse events, including Australia, the Amazon, Siberia, and India, confirming that the proposed multi-agent auction protocol generalizes across different orbital geometries and operational conditions while maintaining efficient, non-redundant task allocation.

The scalability experiment, as shown in Table 3, summarizes the system performance as a function of constellation size N, comparing the greedy baseline with the proposed SwarmSAT controller. For the greedy approach, increasing N slightly improves utility but also leads to extremely high redundancy ratios (up to 0.9865) and a large number of redundant observations, indicating inefficient task allo-

Table 2. SwarmSAT vs Greedy baseline model at different geographical events.

Metric	**Greedy (mean±std)**	**SwarmSAT (mean)**	**Δ%**	**Notes**
Global Utility	0.4831 ± 0.1949	0.6583	+36.3%	Higher is better
Redundancy Ratio	0.9803 ± 0.0061	0.0000	+100.0%	Lower is better (0 = no redundant pop-up obs.)
Adaptability (s)	5025.0 ± 5478.6	60.0	+98.8%	Lower is better (faster response)
Total Popup Obs.	54.25 ± 16.2	1.0	+98.2%	Targeting efficiency proxy

Scenario (For Greedy Model)	**Utility**	**Redundancy**	**Adaptation (s)**	**#Obs**
Australia Bushfire	0.6976	0.9865	720	74
Amazon Deforestation Fire	0.3403	0.9831	4200	59
Siberia Wildfire	0.5970	0.9722	2220	36
India Cyclone	0.2973	0.9792	12960	48

Table 3. Summary of scalability analysis for SwarmSAT and greedy baseline in Australia Bushfire.

N	**System**	**Utility**	**Redun.**	**Adapt. (s)**	**#Obs**	**#Eligible**
3	Greedy	0.6502	0.9565	780	23	3
5	Greedy	0.6742	0.9730	780	37	5
7	Greedy	0.6742	0.9811	720	53	7
10	Greedy	0.6976	0.9865	720	74	10
10	SwarmSAT	0.7723	0.0000	60	1	N/A

cation. In contrast, SwarmSAT maintains competitive utility while completely eliminating redundancy and significantly improving adaptability, reducing response time from hundreds of seconds to only 60 s. Notably, SwarmSAT assigns exactly one satellite to the pop-up target regardless of constellation size, demonstrating that the sequential handoff auction protocol scales effectively while preserving efficient, non-redundant coordination. More ablation studies about SwarmSAT can be found in Appendix (B).

5. Conclusion

This work introduced SwarmSAT, a tool-grounded multi-agent coordination framework for autonomous satellite task allocation under dynamic pop-up events and operational constraints. By combining deterministic auction logic with LLM-driven agent reasoning and a shared, auditable environment state, the proposed system enables transparent, physically grounded, and constraint-compliant decision making. The sequential handoff auction protocol ensures that only one optimal satellite is assigned to each pop-up target, eliminating redundant observations while preserving interpretability and reproducibility. Extensive simulation results across geographically diverse disaster scenarios demonstrate that SwarmSAT significantly improves coordination efficiency compared to a greedy baseline, achieving higher utility, eliminating redundant observations, and reducing response latency from thousands of seconds to near-real-time assignment. Furthermore, scalability experiments confirm that the framework maintains effective coordination as constellation size increases, highlighting its suitability for future large-scale satellite swarms.

Overall, this work demonstrates that LLM-enabled multi-agent systems, when grounded in structured environmental tools and deterministic execution policies, can provide reliable and scalable autonomy for spaceborne sensing missions. Future work will extend this framework to multi-target auctions, heterogeneous sensor capabilities, and real-time deployment using live orbital data, advancing toward fully autonomous, resilient Earth observation constellations.

References

[1] Mitchell Ai-Chang, John Bresina, Len Charest, Adam Chase, JC-J Hsu, Ari Jonsson, Bob Kanefsky, Paul Morris, Kanna Rajan, Jeffrey Yglesias, et al. Mapgen: mixed-initiative planning and scheduling for the mars exploration rover mission. *IEEE Intelligent Systems*, 19(1):8–12, 2004. 2

[2] Dimitri P Bertsekas. The auction algorithm: A distributed relaxation method for the assignment problem. *Annals of operations research*, 14(1):105–123, 1988. 2

[3] Steve Chien, Rob Sherwood, Daniel Tran, Benjamin Cichy, Gregg Rabideau, Rebecca Castano, Ashley Davies, Rachel Lee, Dan Mandl, Stuart Frye, et al. The eo-1 autonomous science agent. In *Autonomous Agents and Multiagent Systems, International Joint Conference on*, pages 420–427. IEEE Computer Society, 2004. 2

[4] M Bernardine Dias, Robert Zlot, Nidhi Kalra, and Anthony Stentz. Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7):1257–1270, 2006. 2

[5] Benedetta Ferrari, Jean-François Cordeau, Maxence Delorme, Manuel Iori, and Roberto Orosei. Satellite scheduling problems: A survey of applications in earth and outer space observation. *Computers & Operations Research*, 173: 106875, 2025. 1

[6] Jeremy Frank, Ari Jonsson, Robert Morris, David E Smith, and Peter Norvig. Planning and scheduling for fleets of earth observing satellites. In *International Symposium on Artificial Intelligence, Robotics, Automation and Space*, 2001. 1

[7] Brian P Gerkey and Maja J Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International journal of robotics research*, 23(9):939–954, 2004. 2

[8] Al Globus, James Crawford, Jason Lohn, and Anna Pryor. Scheduling earth observing satellites with evolutionary algorithms. In *International conference on space mission challenges for information technology*, 2003. 1

[9] Ehud Karpas, Omri Abend, Yonatan Belinkov, Barak Lenz, Opher Lieber, Nir Ratner, Yoav Shoham, Hofit Bata, Yoav Levine, Kevin Leyton-Brown, et al. Mrkl systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning. *arXiv preprint arXiv:2205.00445*, 2022. 2

[10] T Kelso. Norad two-line element set format. *URL https://bit.ly/2DqmGqe*, 2022. 2

[11] T Kelso, Felix Hoots, and Ronald Roehrich. Spacetrack report no. 3-models for propagation of norad element sets. *NASA, Tech. Rep*, 1988. 2

[12] S Knight, Gregg Rabideau, Steve Chien, Barbara Engelhardt, and Rob Sherwood. Casper: Space exploration through continuous planning. *IEEE Intelligent Systems*, 16(5):70–75, 2001. 2

[13] Zhang Lei. Langgraph vs langchain: Which framework is best for ai agents? *MetaVision Journal of Multidisciplinary Studies (MVJMS)*, 1(4):16–26, 2024. 2

[14] Tuomas Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial intelligence*, 135 (1-2):1–54, 2002. 2

[15] Ranjan Sapkota, Rashik Shrestha, Madhav Rijal, and Manoj Karkee. Langchain vs. langgraph vs. langsmith: Taxonomies of agentic ai toolchains for end-to-end orchestration. *Authorea Preprints*, 2025. 2

[16] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023. 2

[17] Benjamin D Smith, Barbara E Engelhardt, and Darren H Mutz. The radarsat-mamm automated mission planner. *AI Magazine*, 23(2):25–25, 2002. 1

[18] David Vallado, Paul Crawford, Ricahrd Hujsak, and Thomas Sean Kelso. Revisiting spacetrack report# 3. In *AIAA/AAS astrodynamics specialist conference and exhibit*, page 6753, 2006. 2

[19] Alpo Van Der Knaap. Modular monitoring and reporting system of satellite ground station. 2024. 2

[20] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37, 1961. 2

[21] 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*, 2022. 2

SwarmSAT Agent: A Tool-Grounded Autonomous Multi-Agent LLM Framework for Rapid Satellite Constellation Retasking of Pop-Up Events

Supplementary Material

A. Data Sources and Simulation Configuration

This section provides detailed information on the data inputs, structured records, and simulation parameters used throughout the SwarmSAT evaluation pipeline. These details complement the main text by explicitly defining the physical orbital inputs, agent-accessible state variables, simulator-derived geometric quantities, and configuration settings that govern visibility computation, bid generation, and coordinated task allocation. Because the proposed framework relies on tool-grounded reasoning rather than purely symbolic planning, each agent decision is conditioned on physically meaningful state information derived from orbital propagation and resource models. The following tables summarize the data components and experimental parameters to ensure full transparency, reproducibility, and auditability of all reported results.

Table 4 describes the structured data inputs and runtime records that support the multi-agent coordination framework. Satellite orbital inputs are derived from two-line element (TLE) sets, which are propagated using standard orbital mechanics models to generate satellite positions, ground tracks, and time-varying visibility geometry relative to ground targets. These propagation outputs form the physical foundation for determining when observations are geometrically feasible. The static target catalog defines routine observation locations and pre-designated critical regions, while dynamically injected pop-up targets simulate urgent disaster events requiring immediate replanning. Simulator-derived visibility window records convert continuous orbital motion into discrete observation opportunities, including elevation angles, nadir angles, and time bounds for each potential observation. Satellite state trajectories, including battery levels, observation timestamps, and resource consumption, provide the resource-aware inputs required for bid computation and eligibility filtering. Finally, negotiation and execution logs capture all agent interactions, including bid submissions, assignment decisions, tool queries, and observation confirmations. These logs provide a complete provenance record of the decision process, enabling post-hoc verification, metric computation, and analysis of coordination behavior.

Table 5 summarizes the simulation configuration used consistently across all experiments. The simulated constellation consists of 10 low Earth orbit satellites whose orbital parameters are initialized from real TLE data. The simulation is executed over a six-hour operational horizon with a temporal resolution of one minute, allowing accurate reconstruction of visibility windows and agent decision timing. Observation feasibility is constrained by a minimum elevation angle of 10°, ensuring that only physically realistic observation opportunities are considered. Satellite resource dynamics are modeled through a battery consumption and recovery process, where each observation incurs a fixed energy cost and solar exposure enables gradual recharge over time. The bid function integrates viewing geometry, represented by the best achievable nadir angle within a visibility window, and the satellite's current battery level, producing a resource-aware priority score used for deterministic winner selection. An urgency weighting term further incorporates temporal availability to prioritize time-sensitive observation opportunities.

All agents operate using a deterministic LLM configuration with fixed inference parameters, ensuring that decision behavior is fully reproducible across repeated simulation runs. Sequential tool execution enforces the ordered handoff structure of the auction protocol, preventing race conditions and ensuring consistent state synchronization between agents and the shared environment. The simulator records all intermediate state transitions, allowing reconstruction of agent reasoning inputs and assignment outcomes. Collectively, these configuration choices establish a controlled, physically grounded, and fully auditable experimental environment for evaluating the performance, robustness, and scalability of multi-agent satellite coordination under dynamic operational conditions.

B. Additional Scenario Visualizations and Constraint Analysis

In this section, we provide additional visualizations that illustrate the geometric context, constraint compliance behavior, and stress-test performance of the greedy baseline coordination framework. These figures complement the quantitative metrics reported in the main text by providing spatial, operational, and system-level perspectives on agent decision-making and robustness under realistic orbital and mission conditions.

Figure 4 shows the orbital geometry and visibility footprint for the Australia Bushfire pop-up event. The satellite ground tracks and visibility windows illustrate how observation opportunities emerge as satellites traverse their orbital paths. The winning satellite (SAT_5) is highlighted at the point of its optimal visibility window, demonstrating how the sequential handoff auction selects the satellite with the most favorable viewing geometry and resource state.

Table 4. Comprehensive details about the data format and role used in this study.

Data component	Format / source	Role in pipeline
Satellite orbital inputs	TLE pairs (external feed or embedded fallback set)	Defines constellation ephemerides for propagation, visibility windows, and ground tracks
Target catalog (static)	CSV / generated target list (routine + critical)	Provides persistent observation demand with lat/lon and priorities
Pop-up events (dynamic)	Procedurally generated runtime targets	Injects urgent, time-critical events for online replanning and coordination tests
Visibility windows	Simulator-derived structured records	Determines eligibility and geometric quality (angles/elevation) for bidding
Satellite state trajectories	Simulator state (battery, timestamps, observations)	Supplies resource-aware inputs to agents and bid computation
Negotiation & execution logs	JSON traces (bids, assignments, transcripts, observations)	Supports metric computation, ablation analysis, and auditability of decisions

Table 5. SwarmSAT's hyperparameter details.

Parameter	Value	Unit	Notes
Constellation size	10	satellites	LEO satellites (Planet Dove TLEs)
Simulation duration	6.0	hours	Total scenario window
Time resolution	1.0	min/step	Visibility sampling interval
Routine targets	100	cities	World-city observation targets
Critical targets	4	zones	Pre-assigned disaster zones
Popup target	1	target	Injected at sim step 30
Min. elevation threshold	10.0	°	Below threshold = not observable
Observation battery cost	5.0	%/photo	Fixed cost per photo
Battery recovery rate	1.0	%/min	Always-on solar recovery
Bid formula	$(1-\theta/90)\times(B/100)$	–	θ = best nadir angle, B = battery %
Urgency weight	$1-t_w/90$	–	t_w = minutes until window opens
LLM model	GPT-5.2	–	Used for all agents
LLM temperature	0.0	–	Deterministic inference
Parallel tool calls	False	–	Disabled: sequential handoffs
Random seed	42	–	For reproducibility

This visualization provides spatial intuition for how geometric constraints directly influence bid computation and task assignment.

Moreover, as can be seen in Figure 5, it summarizes constraint compliance across multiple operational policies, including geographic exclusion zones, satellite-specific restrictions, quality thresholds, and minimum battery requirements. The SwarmSAT controller satisfies all evaluated constraints, achieving full compliance across all scenarios, while the greedy baseline violates several rules. This result confirms that the tool-grounded multi-agent protocol reliably enforces operator-defined policies while maintaining effective task allocation.

For scalability concerns, Figure 6 presents system performance under simultaneous pop-up events, highlighting the effects of increasing coordination load. While coverage remains high for up to three concurrent events, mean adaptability time increases substantially as the number of simultaneous pop-ups grows, reflecting the inherent scheduling complexity under higher demand. These results demonstrate that the system degrades gracefully under stress while preserving core coordination functionality, supporting its applicability to dynamic, real-world disaster monitoring scenarios.

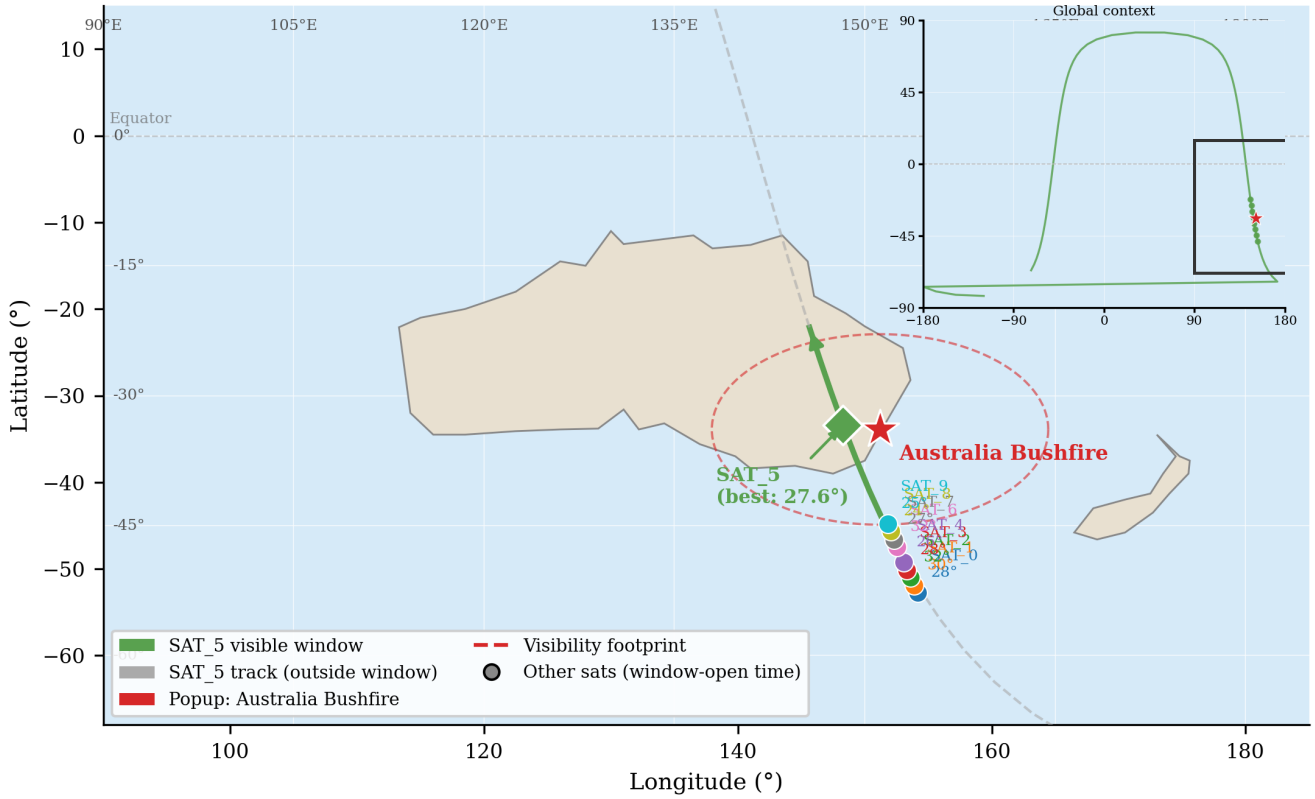

Figure 4. Orbital geometry and visibility footprint for the Australia Bushfire pop-up event. The winning satellite is selected based on optimal viewing geometry and resource state within its visibility window.

To be more transparent about the detailed mechanism that occurred through SwarmSAT, Figure 7 illustrates how the SwarmSAT coordination framework incorporates operator-defined constraints directly into the multi-agent negotiation and task assignment process. When an urgent pop-up event is detected, the Cluster Lead agent initiates an auction and sequentially queries eligible satellites. Each satellite evaluates its local operational state, including battery level, observation geometry, and time to visibility, and determines whether it can safely and legally participate. If a constraint would be violated, the satellite abstains from bidding; otherwise, it submits a bid reflecting its expected observation quality and resource cost. The Cluster Lead then collects all valid bids and assigns the task only if the assignment satisfies all active constraints.

As a complement to Table 1, the schematic highlights several representative constraint categories. First, resource constraints, such as minimum battery requirements, prevent satellites with insufficient energy from participating, ensuring mission sustainability. Second, quality constraints, such as maximum allowable view angle, ensure that only satellites capable of producing sufficiently high-quality observations are considered. Third, operational exclusion constraints allow specific satellites to be reserved for higher-priority missions, preventing them from being assigned to opportunistic tasks. Finally, geographic policy constraints, such as restricted observation zones, may prohibit assignment entirely, in which case the coordinator terminates the auction without selecting a winner.

The Figure 7 also emphasizes the distributed decision-making structure of the protocol. Constraint evaluation occurs locally at each satellite and globally at the coordinator level, ensuring that both individual resource limits and system-wide policies are enforced before assignment. As a result, the final task allocation reflects both optimal observation utility and strict compliance with mission rules. This constraint-aware negotiation mechanism enables autonomous satellite swarms to respond rapidly to dynamic events while maintaining safety, operational integrity, and adherence to mission policies.

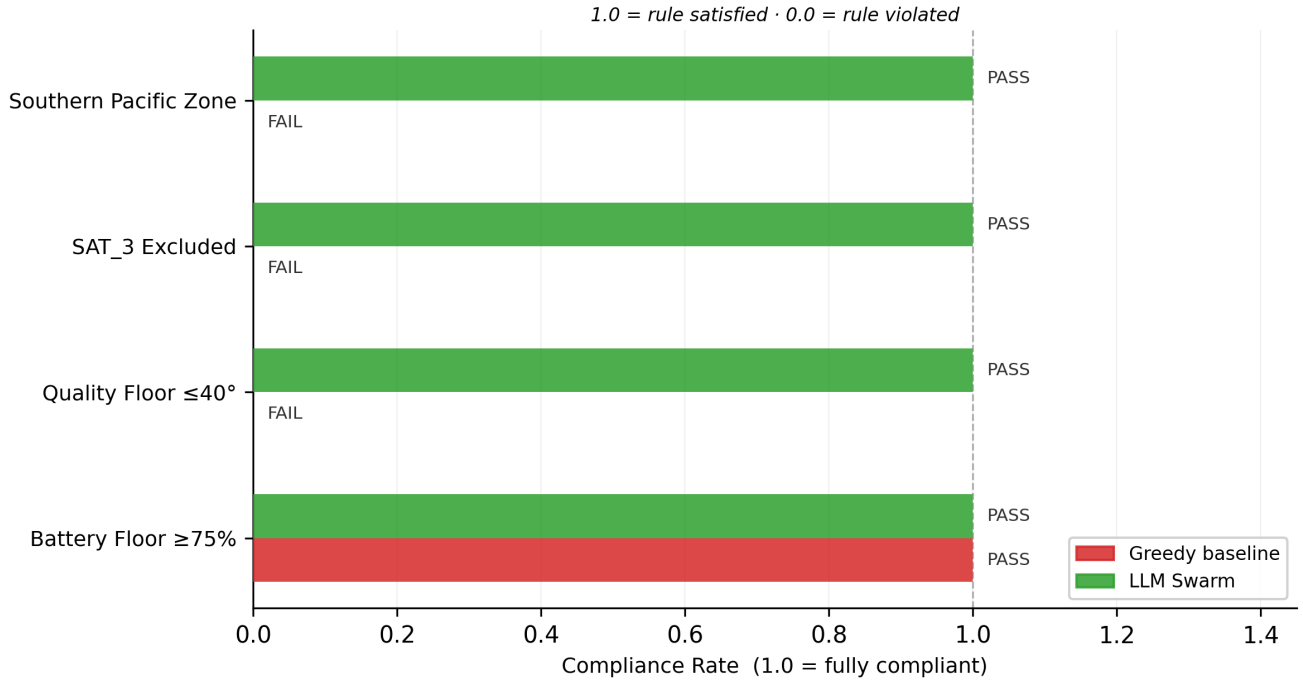

Figure 5. Constraint compliance comparison between the greedy baseline and SwarmSAT.

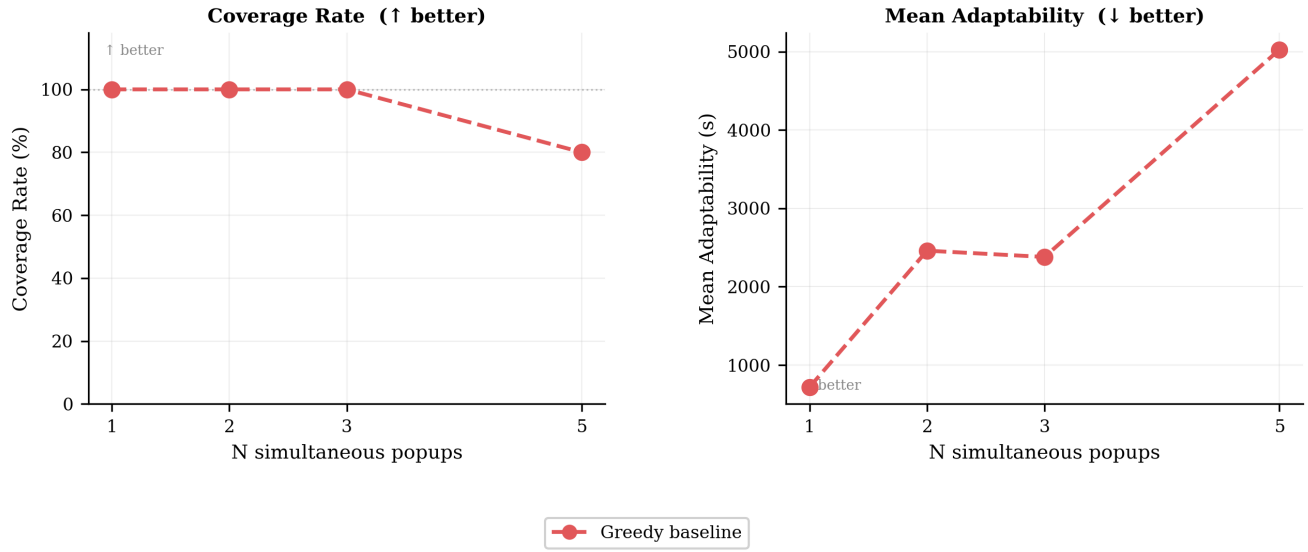

Figure 6. Multi-pop-up stress test results for the greedy baseline.

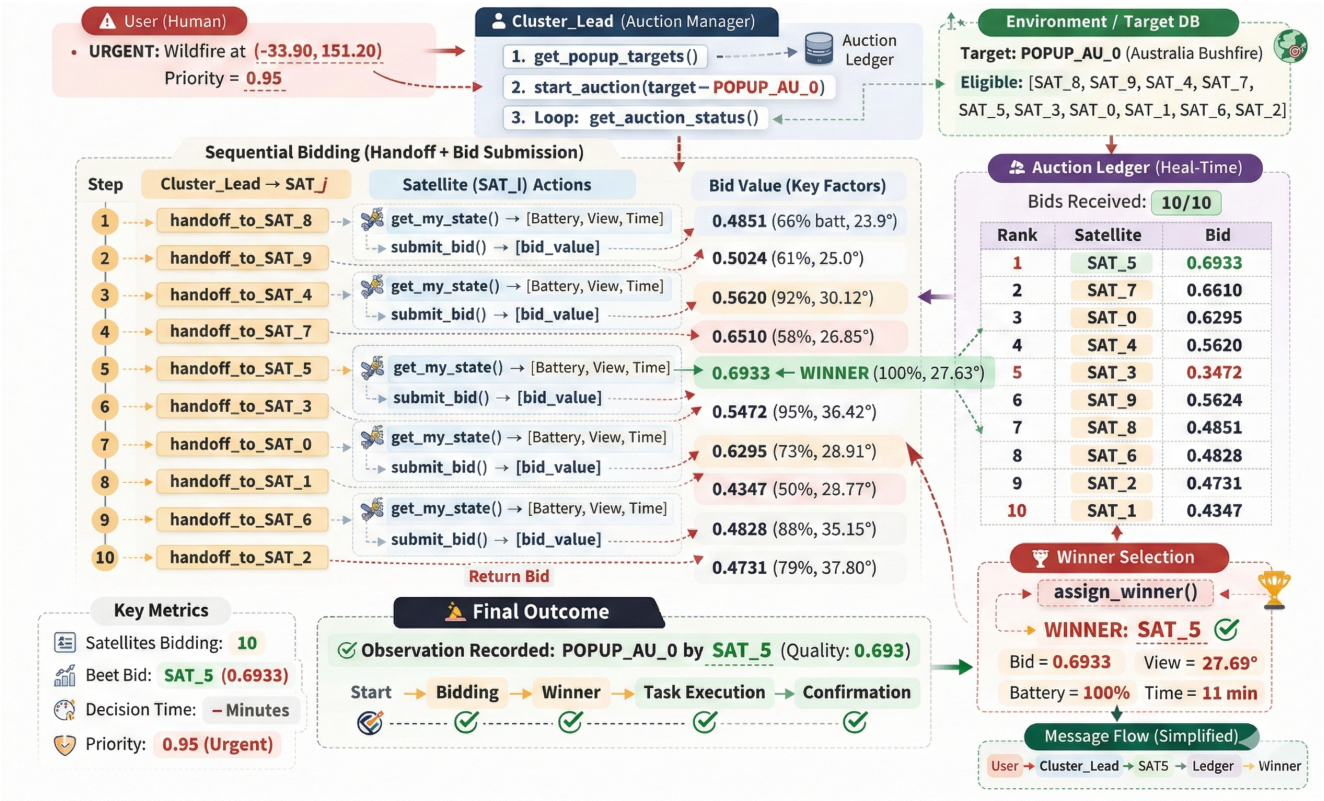

Figure 7. Constraint-aware multi-agent negotiation and task assignment.