

Evidence-Bounded Incident Observability for Resource-Constrained Spacecraft

Nomos Research

hello@nomos.run <https://nomos.run/research/incident-capture/>

NOMOS-TR-2026-03, version 1.1 25 August 2026

Preprint / technical report. Not peer reviewed.

Abstract

Satellite anomaly detection and satellite incident diagnosis are often discussed as one problem. They are not. A detector can identify an unusual telemetry interval without retaining the command, software event, resource transition, or last successful operation needed to explain it. This technical note evaluates a narrower architecture: detect a symptom, preserve prioritized evidence near the affected workload, and defer interpretation to a bounded, human-visible path. Three ground studies provide the evidence. First, a transparent logistic baseline trained on public OPS-SAT-AD features reaches 0.870 F1, 0.957 precision, and 0.796 recall on the untouched 529-segment test split. It finds 90 of 113 labelled anomalies with four false alerts, but the dataset cannot support root-cause claims. Second, on a physical 8 GB Jetson Orin development board, we make one Earth-observation output symptom arise from memory exhaustion, a process deadlock, and storage exhaustion. All 15 injected incidents are detected; a priority-ordered capsule of 380–494 bytes preserves direct cause-specific evidence in 15/15 trials, versus 5/15 for an application-log tail and 0/15 for uniform telemetry at the same 512-byte limit. Ten paired clean controls produce no incidents, and all 15 isolated workloads recover. Third, four quantized open-model packages produce 420 ground-host responses across a bounded-label protocol, 15 intact/redacted evidence pairs, and a retrieved-runbook repeat. The best candidate passes 5/15 pairs without the runbook and 6/15 with it; none clears the pre-specified 15/15 gate, so none advances to Jetson measurement. These results support a deterministic evidence and validation layer around optional models, not an autonomous spacecraft agent. They do not establish flight readiness, commercial operator value, or safe command authority.

0.870

held-out OPS-SAT F1

15/15

512-byte capsules decisive

0/4

model packages clear the

grounding gate

1 Introduction

Low-Earth-orbit spacecraft combine sparse contact windows, constrained downlink, long-lived software, and hardware that cannot be physically inspected. When an onboard workload stops producing output, an operator may receive the symptom long after the causal evidence has scrolled out of a log or been overwritten by nominal telemetry. More downlink does not automatically solve this: the useful bytes may never have been

recorded, or may be buried in a larger stream.

Machine learning has improved telemetry anomaly detection, and recent work explores language-model agents for satellite operations [1, 2, 3, 4]. These systems motivate a question that is easy to skip: what evidence is available to the model after a detector fires? A plausible explanation generated from thin context remains plausible, not causal. Adding onboard inference can reduce response latency, but it does not manufacture missing provenance.

We study an evidence-bounded incident path. The core unit is a small, priority-ordered *incident capsule* containing the trigger, direct system events, last known-good progress, and a short state snapshot. Its purpose is to survive a strict retrieval budget and make unsupported specificity visible. An operator or ground-side agent may interpret it; the recorder does not issue spacecraft commands.

The contribution is an experimental decomposition, not a claim of first agentic observability. We ask three questions:

1. Can a transparent detector find labelled anomalies on held-out real satellite telemetry?
2. Under one external symptom and a fixed byte budget, does prioritized incident evidence preserve cause-specific proof better than familiar retrieval baselines?
3. Can small local language-model packages use decisive evidence and abstain when that proof is removed?

The studies intentionally stop before command recommendation. They also separate two evidence domains: OPS-SAT-AD measures telemetry anomaly detection, while controlled Jetson faults measure cause-specific incident capture. Combining their labels would create a false end-to-end claim.

2 Related Work and System Boundary

2.1 Detection and operations tooling

Spacecraft anomaly detection spans supervised, unsupervised, and hybrid methods. Hundman et al. combine LSTMs with dynamic thresholding on spacecraft telemetry [1]. OPS-SAT-AD provides a public benchmark derived from ESA OPS-SAT telemetry, with official training and held-out test segments [2]. At operational scale, DLR’s ATHMoS reports three years of ground-side novelty detection on TerraSAR-X and TanDEM-X, emphasizing operator feedback, relabelling, and false-positive control rather than a one-time model score [5].

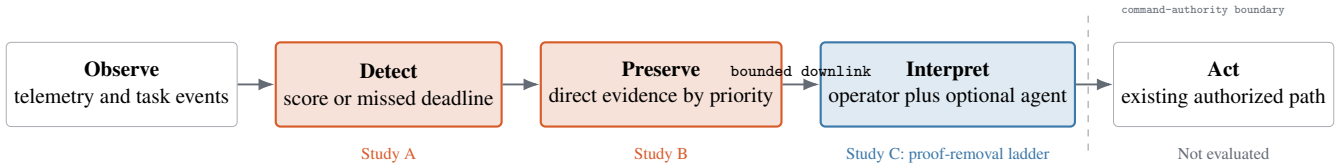


Figure 1: The proposed evidence path separates observation, detection, preservation, interpretation, and action. This report measures the orange stages and four ground-side model packages under an evidence-removal gate. It does not evaluate autonomous command generation or execution.

Detection is only the first transition. Causal-inference work at GSOC explicitly targets explanations rather than anomaly scores [6]. COMET combines an LSTM autoencoder, retrieval, and a Gemma-based agent over simulated multivariate telemetry [3]. Its reported detector trades high recall for lower precision, and its paper identifies grounding and hallucination evaluation as open work. We view these efforts as complementary: a model may direct attention, but direct events and provenance should remain distinguishable from inference.

2.2 Agents, integration, and operator context

DLR has explored local language-model agents over a simulated cFS/YAMCS stack under human-in-the-loop control [4]. EUMETSAT proposes stable interfaces for incrementally adding AI and non-AI agents to existing mission-control systems [7]. Both support an adapter boundary rather than wholesale replacement of flight or ground software.

A KARI-centered operator study found fragmented tools, manual communication, and diagnosis dependent on memory. Its prototype feedback favored a minimally styled, alert-focused view with traceable status changes and a seamless real-time/playback transition [8]. That is evidence for unified context and traceability, not proof that any single interface improves mission outcomes. Our proposed ground view follows those design requirements and retains links from every conclusion to source evidence.

3 Study A: Held-Out Satellite Telemetry

3.1 Dataset and protocol

OPS-SAT-AD contains 2,123 labelled segments from nine OPS-SAT magnetometer and photodiode channels, represented by 18 published handcrafted features [2]. We preserve the official 1,594-segment training and 529-segment test split. The test set contains 113 labelled anomalies. The official test labels remain untouched during model and threshold selection.

Within the training portion, a deterministic split assigns 1,243 segments to fitting and 351 to validation. A regularized logistic classifier is selected over fixed feature transforms, L2 strengths, and positive-class weights. Selection yields a signed-log polynomial transform, L2 coefficient 0.0003, positive weight 1.992988, and validation threshold 0.755. The

Table 1: Held-out OPS-SAT-AD result at the validation-selected 0.755 threshold.

Metric	Value	Metric	Value
Precision	0.957	Recall	0.796
F1	0.870	Accuracy	0.949
AUROC	0.989	Avg. precision	0.965

	Pred. anomaly	Pred. nominal
Actual anomaly	90	23
Actual nominal	4	412

resulting classifier is deliberately transparent and compact; it is a workflow baseline, not a claim that logistic regression is the best OPS-SAT detector.

3.2 Held-out result

Table 1 reports the untouched test result. The detector identifies 90 of 113 anomalies, misses 23, and produces four false alerts among 416 nominal segments. Precision is high, while the missed-anomaly count makes recall the binding weakness. Threshold changes would move this tradeoff; no operator cost function was available to calibrate it.

The selected feature transform and classifier execute on a physical Jetson Orin ground bench. Across 400 vectorized passes over the 529-segment batch, median amortized time is 0.7499 microseconds per segment and p95 is 0.7756 microseconds. This measurement excludes raw waveform feature extraction, I/O, scheduling, and serialization; it is not end-to-end onboard latency. The fitted state would occupy 296 bytes if serialized as float32, while the measured NumPy process uses float64 arrays.

For three displayed cases only—a detected anomaly, a false alert, and a missed anomaly—a compact evidence record containing segment identity, time bounds, score, threshold, decision, top observed features, and an explicit UNKNOWN cause occupies 251–268 bytes. These byte checks do not establish a size distribution over all 529 segments.

3.3 Detection is not diagnosis

These channels omit the power, thermal, command, software-event, mode, and operator context needed for root cause. A high score and feature contribution show what influenced the

classifier, not why the spacecraft entered that state. We preserve `cause=UNKNOWN` for every displayed case. Study A validates detection only.

4 Study B: Physical Incident Capture

4.1 One symptom, three injected causes

We next test whether the evidence path can preserve cause-specific proof under a strict retrieval budget. A TensorRT 10.3 Earth-observation workload runs on a physical 8 GB Jetson Orin development board. Before fault injection, the accepted engine is checked on all 5,400 images of the fixed EuroSAT test split and reaches 90.426

Every trial contains ten jobs at a 150 ms cadence. Job six receives one of three safely isolated faults: (1) a 1 GiB cgroup memory limit produces an out-of-memory kill and exit code 137; (2) a true software deadlock triggers a local stack watchdog at 400 ms; or (3) a 128 KiB test volume returns `ENOSPC` while writing a 32 KiB product. In all cases the external trigger is identical: the expected product has not appeared by 500 ms.

There are five repetitions per cause, 15 fault trials total, plus five counterbalanced clean pairs, or ten clean controls. Fault order rotates across repetitions and no measured trial is removed. Docker, cgroups, and tmpfs provide safe ground isolation; they are not proposed as a flight stack.

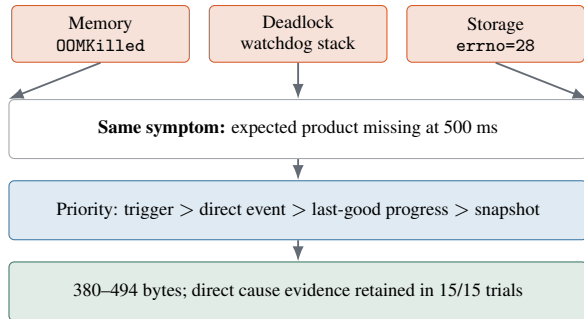


Figure 2: One missing product is caused by three isolated faults. A fixed priority order protects direct evidence before lower-value context is truncated.

4.2 Retrieval formats and evaluator

At each configured byte budget we compare three representations. The *application-log tail* retains the most recent application text. *Uniform telemetry* samples periodic resource observations without incident priority. The *incident capsule* serializes the trigger, direct system event, last successful step, and selected snapshot fields in that order.

A conservative deterministic evaluator returns a cause only when its direct signature survives: the cgroup OOM event, the watchdog stack record, or the `ENOSPC` error. Otherwise it returns `UNKNOWN`. This is a direct-evidence test, not a learned diagnosis score. The evaluator was frozen before the final runs.

4.3 Results

All 15 injected incidents are detected. Median detection latency is 507.79 ms for memory exhaustion, 506.12 ms for deadlock, and 508.46 ms for storage exhaustion. No incident fires in the ten clean controls. Restarting only the isolated inference worker restores a verified product in all 15 fault trials, with median recovery between 1.688 and 1.706 seconds by cause. No spacecraft command or shared subsystem is exercised.

Figure 3 and Table 2 show the retrieval result. At 4 KiB, the application tail preserves the storage error in 5/15 trials, uniform telemetry preserves the memory signal in 5/15, and the capsule preserves all three causes in 15/15. At 512 bytes, uniform telemetry loses its useful context, while the application tail still retains only storage errors. Every capsule remains decisive at 380–494 bytes.

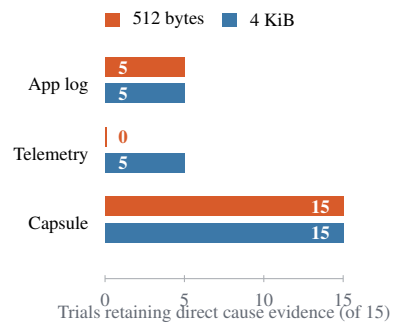


Figure 3: Cause-specific evidence retained under two retrieval budgets. A value of zero means the conservative evaluator abstained, not that it guessed incorrectly.

Table 2: Retrieval outcome by representation. “Coverage” names the cause classes for which all five repetitions remain decisive at 4 KiB.

Representation	512 B	4 KiB	Coverage at 4 KiB
Application tail	5/15	5/15	Storage only
Uniform telemetry	0/15	5/15	Memory only
Incident capsule	15/15	15/15	All three

The complete, untruncated capsules occupy 852–1,061 bytes (median 891). At 512 bytes, lower-priority context is omitted but direct evidence remains. Five paired observation-overhead controls range from -0.448% to $+0.130\%$, with median -0.078% . Because the interval spans zero and the sample is small, the supported statement is that no slowdown was measured at this resolution, not that observation has zero cost.

Three pilot failures changed the protocol. A TensorRT 10.3 plan did not load in the TensorRT 10.9 isolation container, so the pinned ONNX was rebuilt and revalidated on 5,400 images. A process that sent itself `SIGSTOP` was resumed by the container runtime, so those pilot runs were rejected and replaced with a true deadlock. Finally, an unprivileged external observer could not read a useful wait channel, so diagnosis

moved to a bounded local stack watchdog. These divergences are retained in the public evidence package.

5 Study C: Evidence-Grounded Model Ladder

5.1 Protocol and qualification gate

Prioritized evidence is useful only if an interpreter can consume it. An initial Qwen3 0.6B Q6_K sanity check returned UNKNOWN for all 45 fixed-label prompts. It created no unsupported diagnoses but also missed every decisive capsule. Version 1.1 adds a separately versioned candidate ladder and does not combine the two protocols numerically.

The ladder evaluates four quantized, Apache-2.0 model packages: Qwen3 0.6B Q8_0, Granite 3.3 2B Instruct Q4_K_M, Qwen3 1.7B Q8_0, and Qwen3 4B Q4_K_M. These are deployable package comparisons, not a controlled parameter-scaling study: architecture, training, chat template, and quantization differ.

Each package produces 105 outputs across three frozen protocols. First, bounded-label v2 repeats the 45 prompts from Study B: 15 trials in each of the application-tail, uniform-telemetry, and incident-capsule formats. The output is one cause label or UNKNOWN. Second, counterfactual-grounding v1 evaluates the 15 decisive capsules twice. The original contains the direct OOM, watchdog-stack, or ENOSPC record; a pre-specified intervention removes only that record from the paired copy. The model must return the correct cause and cite the decisive record when it exists, then return UNKNOWN with no citations after removal. Third, retrieved-runbook grounding v1 repeats the same 15 pairs while supplying the exact three proof rules used by the deterministic evaluator.

All 420 responses run through 11ama.cpp 0.3.16 on an Apple M5 ground host with eight CPU threads, zero GPU layers, temperature zero, seed seven, and each package’s embedded chat template. Incident identifiers are opaque. The fixed-label protocol allows 12 output tokens; structured protocols allow 64. This is a ground-side grounding benchmark, not an onboard latency or memory measurement.

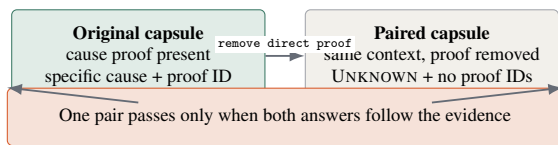


Figure 4: The counterfactual gate tests reliance on evidence, not only answer accuracy. A plausible cause after its sole proof is removed fails the pair.

The Jetson qualification rule was frozen before candidate inference. A package must pass all 15 zero-shot pairs, cite the decisive evidence, and make zero unsupported specific diagnoses in the 45-prompt fixed-label protocol. If more than one package qualifies, the smallest advances. The retrieved-runbook repeat uses the same 15/15 pair rule. No average-score

fallback is permitted.

5.2 Results

Table 3 reports paired grounding and unsupported specificity separately. “BL unsafe” counts specific diagnoses among the 25 fixed-label prompts for which the deterministic evaluator finds insufficient evidence. “ZS unsafe” and “RB unsafe” count specific diagnoses on the 15 proof-redacted prompts. Lower unsafe counts are better; a pair count reaches 15 only when both the intact and redacted answers are correct and properly cited.

No package clears either 15/15 pair gate, so none advances to the Jetson. Without a runbook, Qwen3 4B is the only candidate to pass any complete pairs: 5/15. It still emits ten unsupported specific diagnoses after decisive proof is removed. With the runbook, its unsupported count falls to zero and all 15 redacted prompts produce supported abstentions, but only six intact prompts contain the correct cause and decisive citation. The resulting 6/15 is safer but still operationally incomplete.

Granite demonstrates why output validity is not grounding. It returns strict JSON for all 30 zero-shot structured prompts and 28/30 runbook prompts, yet passes 0/15 and 2/15 pairs respectively because it continues naming causes without proof. Qwen3 1.7B improves to 5/15 with the runbook but retains ten unsupported diagnoses. Qwen3 0.6B remains incapable of completing a pair.

The result is protocol-sensitive and non-monotonic. Package size alone does not explain it, and supplying exact rules changes safety and recall differently. Fine-tuning, constrained decoding, more capable models, broader procedures, and symbolic extraction remain untested. The supported conclusion is narrower: these four packages cannot be trusted to authorize a diagnosis from this evidence, even when one is prompted with the evaluator’s own proof rules.

6 Architecture Implications

6.1 The agent is not the first component

The strongest result in this report is not agentic. A deterministic priority rule preserves direct evidence that ordinary tails and periodic samples discard. Study A shows that detection can be compact and fast after features exist; Study B shows that diagnosis becomes possible only when causal signals are deliberately captured; Study C shows that model size, valid output syntax, and an explicit runbook still do not guarantee evidence-grounded diagnosis.

The resulting product boundary is an incident recorder and evidence layer beside selected onboard workloads. Its minimum responsibilities are to accept task progress and health events, detect a missed contract or consume a detector trigger, snapshot a bounded set of direct evidence, and emit a versioned file, event, or packet through the existing downlink path. It need not replace flight software, the flight computer, mission control, or the operator’s command system.

Table 3: Four-package grounding ladder. Package size is the serialized decimal size. ZS is zero-shot; RB supplies the retrieved proof rules. No package satisfies the pre-specified gate.

Model package	Size (GB)	BL unsafe /25	ZS pairs /15	ZS unsafe /15	RB pairs /15	RB unsafe /15	Qualified
Qwen3 0.6B Q8_0	0.64	0	0	15	0	15	No
Granite 3.3 2B Instruct Q4_K_M	1.55	24	0	15	2	13	No
Qwen3 1.7B Q8_0	1.83	25	0	15	5	10	No
Qwen3 4B Q4_K_M	2.50	0	5	10	6	0	No

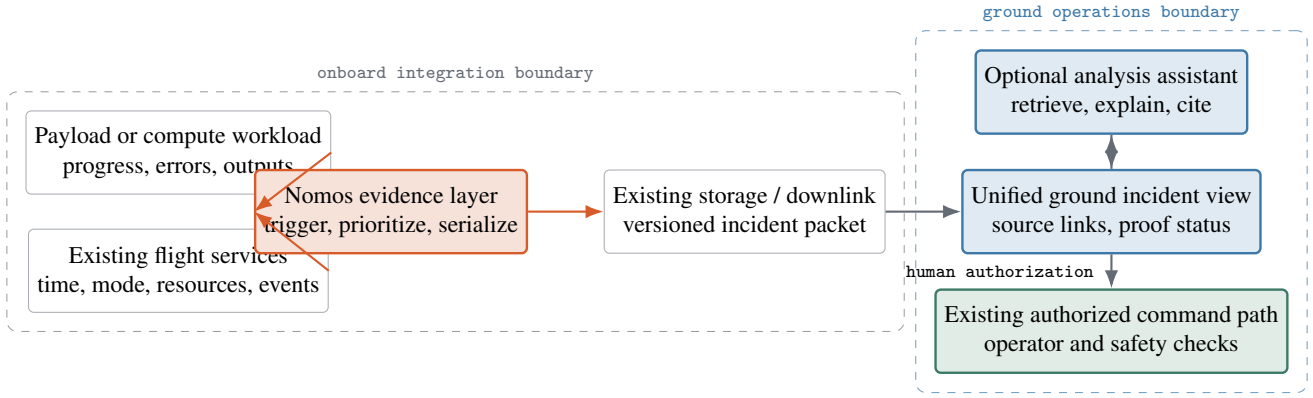


Figure 5: A narrow hybrid architecture. Nomos can integrate beside a workload and export evidence through existing storage and downlink. Deterministic software validates cited proof before a model explanation reaches the operator view. An agent, digital-twin check, or operator workflow does not give the recorder command authority. Interfaces are conceptual and not flight validated.

6.2 Ground-first interpretation and human control

A hybrid deployment is more plausible than placing a general agent in the command path. Onboard logic can capture fleeting evidence and perform cheap, bounded detection while contact is unavailable. Ground systems have more compute, historical context, procedures, model update capacity, and human supervision. A ground assistant can summarize a capsule, retrieve similar incidents, propose missing checks, and expose uncertainty. Deterministic software should reject a specific diagnosis unless its cited record satisfies a versioned proof rule. The operator sees both the model statement and validation result; every accepted claim resolves to the underlying event or telemetry interval.

Digital twins can provide an additional validation boundary for proposed command plans, but simulation acceptance is not proof of a correct diagnosis or safe action. Model fidelity, initial state, and unmodelled couplings remain assumptions. This report evaluates neither a digital twin nor command-plan approval.

The KARI study suggests four interface requirements for the next operator test [8]: one alert-centered incident surface, an explicit chain from status change to evidence, live and replay modes that share the same mental model, and progressive disclosure rather than a dense telemetry wall. Those requirements shaped the companion web workbench, but their value must be validated with commercial LEO operators rather than inferred from a prototype.

7 Threats to Validity

- **No end-to-end mission dataset.** OPS-SAT-AD provides real telemetry labels but not the causal context used in Study B. The Jetson study provides direct causal events but uses controlled faults and public imagery. We do not join them into an artificial diagnosis benchmark.
- **Small controlled sample.** Five repetitions per injected cause and ten clean controls can expose deterministic loss of evidence, but they cannot estimate commercial incident prevalence or rare failure coverage.
- **Engineered fault classes.** Memory, deadlock, and storage exhaustion are useful system faults, not a taxonomy of spacecraft failures. The priority policy may need different evidence for radiation effects, buses, timing, sensors, power, thermal state, or distributed services.
- **Ground hardware only.** The Jetson development board, Linux containers, local watchdog, and test storage are not radiation tolerant or flight qualified. There is no spacecraft power plant, thermal-vacuum environment, attitude history, flight network, or real contact schedule.
- **Measurement boundary.** OPS-SAT timing excludes feature extraction and I/O. The paired overhead result is resolution-limited. Recovery restarts only an isolated workload.
- **Limited model ladder.** Four quantized packages on one ground host do not establish a scaling law or onboard result. Architectures, training, chat templates, and quanti-

zation differ. Fine-tuning, constrained decoding, symbolic extraction, and larger models remain untested.

- **No operator validation.** Commercial LEO operators have not supplied incident data, evidence budgets, workflow baselines, or command constraints for these studies. The experiment shows a mechanism; it does not establish a buyer or operational utility.

8 Reproducibility

The public evidence package contains the 25 physical trial records, 360 regenerated retrieval payloads, source snapshots, manifests, compact summaries, and checksums. The OPS-SAT benchmark script downloads and verifies the published source archive, preserves the official split, performs training-only model selection, and emits the held-out result. A separate weight-free model archive contains all 420 recorded outputs, sanitized inputs, comparison summaries, runner code, package identities, and internal checksums. Model weights are not redistributed. The web workbench presents all three studies in one interface; its selected examples illustrate the aggregate results rather than replacing them.

Table 4: Public artifact identities. Hashes bind the exact files used in this report.

Artifact	SHA-256
OPS-SAT result	be91f3633f7f412efa4f9e9f870860cdd79c9d8b95804def101d5c18857bbdf38fccdea8988997074b97762ff4831f8be5841fff40f45f7a97b4c49e969fbaf5
Jetson summary	bf19e59c0107a7fb3f92c8a
Evidence package	ae528fa6bddf1d8c40a3721cd03a5e4740f4b323326a01405802041e9268de7a0e77d9459a01174d91397a8fb4546cbb6d12da67d
Model comparison	5574ec255dbf040cfe138f83305694a6b47d7c16c3d4f933c2273d56e3f47d15
Model output archive	

The publication page is <https://nomos.run/research/incident-capture/>. One workbench at <https://nomos.run/research/incident-capture/replay/> contains the OPS-SAT, Jetson, and model-grounding views. The OPS-SAT source dataset remains under CC BY 4.0 at DOI 10.5281/zenodo.12588359. Exact environment versions, model revisions and hashes, prompts, limitations, and premeasurement divergences are recorded in the machine-readable summaries.

9 Conclusion and Next Gate

The studies reject a simple end-to-end agent story. OPS-SAT anomalies can be detected without causal context; direct evidence for three controlled compute faults fits in 512 bytes; and none of four open-model packages passes a strict proof-removal gate. Detection, evidence capture, model interpretation, deterministic validation, and operator action remain separate.

The supported architecture preserves evidence near a workload, uses existing storage and downlink, and lets operators or constrained ground tools interpret it traceably. A model may retrieve and explain; deterministic software checks the cited proof; a human remains responsible for the next operational step. Command authority remains outside.

Next, commercial LEO teams should replay recent incidents against the capsule: which decisive signal disappeared, when was it available, what could have been captured, and what byte or contact limit applied? A useful pilot reconstructs one real incident faster or with fewer contacts than today’s workflow. Until then, this is a ground mechanism, not validated spacecraft observability.

References

- [1] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 387–395, 2018.
- [2] Bogdan Ruszczak, Krzysztof Kotowski, David Evans, and Jakub Nalepa. The OPS-SAT benchmark for detecting anomalies in satellite telemetry. *Scientific Data*, 12:710, 2025.
- [3] Haley Calderon, Vaibhav Bhosale, Ketan Bhardwaj, and Ada Gavrilovska. Toward trainable AI agents for satellite operations. In *Small Satellite Conference*, 2026.
- [4] Michele Campanelli and David Hiebl. Conceptual use and architecture of LLM-agents in satellite operations. In *SpaceOps 2025*, 2025.
- [5] Kathrin Helmsauer, Agnese Del Moro, Tobias Goettfert, Clemens Schefels, and Leonard Schlag. AI-based novelty detection in space operations: Three years of operational experience and progression at GSOC. In *SpaceOps 2025*, 2025.
- [6] Clemens Schefels, Bilel Ben Salem, Andreas Gerhardus, Kathrin Helmsauer, Baptiste Lambert, Julia Niebling, Oana-Iuliana Popescu, Martin Rabel, Ferdinand Rewicki, and Leonard Schlag. Explaining satellite anomalies: Causal inference for space operations. In *SpaceOps 2025*, 2025.
- [7] José Feiteirinha, Diogo Vicente, and Luigi Palladino. A scalable architecture for integrating AI and non-AI agents in space mission control applications. In *SpaceOps 2025*, 2025.
- [8] Youeyun Jung, Jaedong Seong, Saehan Song, and Okchul Jung. Integrating digital twins in space operations: Challenges and solutions. In *SpaceOps 2025*, 2025.