

Beyond DVFS: Queue-Stable Model-Tier Selection for Resource-Constrained Onboard Inference

Nomos Research

hello@nomos.run <https://nomos.run/research/beyond-dvfs/>

NOMOS-TR-2026-01, version 1.0 24 August 2026

Preprint / technical report. Not peer reviewed.

Abstract

Hardware clock scaling can reduce instantaneous power during onboard inference, but it cannot change a model’s error profile or service demand. We ask when a software controller that can also select among model tiers adds value over application-aware dynamic voltage and frequency scaling (DVFS). We train and screen three dense CloudScout width tiers, adapt them to Landsat-9 using 16 unlabeled target tiles per acquisition fold, compile FP16 TensorRT engines, and measure 12 model-frequency points on a Jetson Orin Nano in its 15 W configuration. The full tier is dominated; medium and tiny form a measured two-mode frontier. We then compare best-static, fixed-schedule, adaptive-DVFS, and joint model/DVFS policies over a frozen matrix of four synthetic resource traces, three arrival rates, five error-cost ratios, and the same 320 real Landsat-9 tiles per scenario. A first controller completed all tasks in its smoke cell but switched model 24 times; we stopped the matrix and preserved the failure. A separately frozen revision admits only service-stable actions when one exists and uses asymmetric promotion hysteresis. It reduced the smoke to two transitions. Across 60 deterministic scenario cells, joint selection beat adaptive DVFS on normalized mission utility in 12, tied in 48, and lost in none. Every gain required a high false-clear cost and near-saturation or overload; every underloaded and lower-cost cell tied. Frequencies below 612 MHz are profile-replayed, traces and thermal dynamics are synthetic, loss ratios are not operator-calibrated, and scenario cells reuse one shard. The result is therefore a sensitivity boundary and controller failure study, not statistical, spacecraft, or flight evidence.

1 Introduction

Onboard inference can discard cloud-obscured imagery, prioritize events, or compress a payload before downlink. CloudScout is one concrete example: a compact convolutional network intended for cloud detection at the spacecraft edge [1]. Such workloads live inside a power, thermal, memory, and timing envelope. The usual software-visible response is to start or stop the workload, while the hardware governor changes clocks to stay within a power mode. Neither operation asks whether a different model would preserve more mission value during the constrained interval.

The distinction matters because clock and model controls act on different axes. DVFS changes latency and power for

one fixed error function. A model change alters latency, energy per item, and the relative rates of false clears and false clouds. A smaller model is not automatically less useful, and a lower clock is not automatically more energy efficient. The controller must reason about a measured action frontier, workload arrival rate, deadlines, resource state, and a mission-specific loss ratio.

This report tests that proposition on a development board before claiming relevance to a spacecraft. It makes three contributions:

1. A measured quality/resource screen for three CloudScout width tiers at four GPU frequencies. It identifies two non-dominated modes and shows that, for all tiers, the highest tested clock minimizes whole-board joules per tile.
2. A four-policy compute-in-the-loop (compute-HIL) harness with sealed scoring, common safety logic, and a fair adaptive-DVFS baseline. Its first use exposes a deterministic model-switch limit cycle, causing the planned matrix to stop.
3. A predeclared queue-stability invariant and a 60-cell sensitivity map that identifies where model choice adds utility beyond DVFS—and, equally important, 48 cells where it does not.

The narrow conclusion is that software-level model selection can add value when at least two measured modes are useful, the preferred tier cannot sustain a binding interval, and the cost of missing work is high. It is not evidence that a large “model zoo” should be carried, that live control beats an orbit-aware schedule, or that this controller is ready for flight.

2 Related Work

2.1 Onboard Earth-observation inference

CloudScout demonstrated a small CNN for on-board hyperspectral cloud detection [1]. Du et al. later studied sensor-domain transfer for satellite-borne multispectral cloud detection and tested bandwidth-efficient supervised and test-time adaptation methods on deployable accelerators [2]. Our model and data lineage follows that public implementation, while our prior cross-sensor pilot tightens the target split by parent acquisition. We use dynamic unsupervised adaptation (DUA), which updates batch-normalization statistics without backpropagation [3], to create target-adapted inference tiers. Sensor adaptation motivates the mission loss, but the controller evaluated here does not adapt weights online.

2.2 Adaptive neural inference

Slimmable networks adjust width at runtime [4]; once-for-all networks train a large family of deployable subnetworks for different hardware budgets [5]. Early-exit architectures such as BranchyNet vary computation by input difficulty [6]. E4 co-optimizes early exits and DVFS for edge video inference [7]. These works establish that accuracy/latency tradeoffs can be exposed to a runtime. Our experiment uses independently compiled, already-resident dense engines rather than a shared slimmable graph or per-input confidence exit. Its focus is controller behavior under changing resource caps and arrival rates, with a fixed aggregate quality prior.

Real-time scheduling theory separates execution demand from deadline feasibility [8]. Our queue-stability rule is deliberately simple: when a sustainable action exists, the controller must not consume accumulated slack with an action whose p95 service time exceeds a fixed fraction of the arrival interval. This is an engineering invariant, not a new scheduling theorem.

3 Problem Formulation

3.1 Actions, loss, and utility

A stream contains N binary cloud decisions. Task i arrives at a_i with deadline d_i . An action $x = (m, f)$ chooses model tier m and GPU frequency f . The frozen hardware table provides p95 service time s_x , mean and peak whole-board input power, and energy per tile e_x .

For model m , let r_m be cloud recall and q_m its false-cloud rate. With assumed cloud prevalence $\pi = 0.5$ and false-clear:false-cloud cost ratio c , the controller’s aggregate expected classification loss is

$$L_m(c) = \pi c(1 - r_m) + (1 - \pi)q_m. \quad (1)$$

The policy sees these frozen priors, never a task label or per-tile reference prediction. A missed or shed task incurs $\max(1, c)$. After hardware execution, a separate scorer joins the sealed answers and reports

$$U = 1 - \frac{\sum_{i=1}^N \ell_i}{N \max(1, c)}, \quad (2)$$

where correct, false-cloud, and false-clear losses are 0, 1, and c . This normalized utility is a sensitivity instrument, not a priced mission objective.

3.2 Safety feasibility and service stability

All policies observe the same synthetic payload state: remaining energy, one-node temperature, current power cap, queue clock, and the residual between nominal and observed resource traces. The common planner rejects actions that exceed the observed peak-power cap, projected energy reserve, 70°C synthetic thermal limit, or task deadline. The thermal plant is a lumped RC model with $R = 2^\circ\text{C}/\text{W}$ and $C = 4 \text{ J}/^\circ\text{C}$; it is not fitted to a spacecraft or the Jetson enclosure.

Let $\Delta_i = a_i - a_{i-1}$ be the current inter-arrival interval. Version 2 marks a feasible action service-stable when

$$s_x \leq \rho \Delta_i, \quad \rho = 0.95. \quad (3)$$

If at least one allowed feasible action is stable, every adaptive policy selects only from that set. If a fixed-model policy has no stable action, it retains the common feasible partial-admission behavior; DVFS is not disabled to favor joint selection.

For joint selection, an unstable current model may demote immediately. Promotion to a lower-expected-loss model requires eight consecutive observations for which the candidate remains stable and at least one arrival interval of projected deadline slack. Model-promotion evidence and the four-task frequency dwell are tracked separately. Safety and stability override dwell.

3.3 Compared policies

The policies differ only in which action dimensions may change:

Best static. One model-frequency pair selected from the nominal trace before the run.

Fixed schedule. A complete nominal action schedule computed before the run; it cannot replan from an unannounced disturbance.

Adaptive DVFS. The expected-loss-minimizing model is frozen before the run, while frequency adapts from live state through the common planner.

Joint (Nomos). Both model and frequency may adapt from the same observed state and action table.

Within each scenario, all policies receive the same task order, nominal trace, initial state, quality priors, action profiles, predictor horizon, safety filter, and scoring rule. There is no post-hoc baseline selection.

4 Experimental Method

4.1 Model tiers and sensor transfer

The upstream full CloudScout accepts three 512×512 bands and contains 1,292,546 parameters. We construct independent dense $0.5 \times$ and $0.25 \times$ width networks by selecting structured channels from the pinned full checkpoint using batch-normalization scale and final-layer input norm. These smaller networks contain 326,786 and 83,522 parameters. They are fine-tuned on a parent-acquisition-disjoint Sentinel-2 split with inverse-frequency weighted cross entropy; checkpoint selection uses only source validation balanced accuracy.

For target screening, each of five Landsat-9 folds uses four parent acquisitions for unlabeled calibration and the other 16 for testing. DUA updates only batch-normalization running statistics from 16 label-blind, hash-selected calibration tiles, then freezes the model. The folds are blocked sensitivity checks, not independent datasets: their test sets overlap, and

Table 1: Tier quality and resource measurements at 612 MHz. Quality is the mean of five acquisition-blocked Landsat-9 DUA folds. Power is Jetson rail telemetry; idle power is not subtracted.

Tier	Params	Bal. acc.	Cloud rec.	False cloud	p95 ms	J/tile
Full	1,292,546	0.8006	0.7485	0.1474	13.431	0.14074
Medium	326,786	0.8089	0.7906	0.1727	5.248	0.05615
Tiny	83,522	0.8223	0.7650	0.1204	2.538	0.02547

the same locked target population later supplies the controller quality priors and scoring shard.

4.2 Orin execution and profiling

The target is a Jetson Orin Nano development system in NVIDIA’s 15 W power mode [9]. Every tier is profiled for 30 seconds at 306, 408, 510, and 612 MHz with 100 ms telemetry. Engines use TensorRT 10.9 FP16 compute and FP32 I/O. The resource sweep uses source-weight engines with the same graphs as the adapted engines and one repetition per cell. Jetson VDD_IN telemetry is not an external calibration-grade meter.

Adapted medium and tiny engines each preserve 32/32 PyTorch decisions on a locked parity shard, with maximum cloud-probability differences of 0.00371 and 0.00418. Both execution contexts remain resident. A separate full/medium screen found resident alternation differed from blocked execution by -0.071% , within measurement noise, with 2,518 MB peak system memory. Cold engine construction is excluded.

Figure 1 shows two important screens. First, tiny dominates full, so full is excluded rather than retained as a weak baseline. Second, downclocking decreases instantaneous board power but increases energy per completed tile in every tier. DVFS remains useful for obeying a peak-power cap, but it is not the energy-minimizing lever in this workload.

4.3 Task shard and scenario matrix

A label-blind hash selects 20 real Landsat-9 tiles from each of 16 held-out parent acquisitions, giving 320 tasks. The controller manifest contains image names and checksums but no labels or reference predictions. The answer key remains off the Jetson; the runner refuses to start if it is present.

The frozen matrix crosses four resource traces, three workloads, five cost ratios, and four policies. This yields 60 scenario cells and 240 hardware policy runs. Workload arrival/deadline pairs are 12/24 ms (under), 6/12 ms (near), and 3.5/7 ms (over). The power-cap transient changes from 12.0 to 9.1 W for tasks 96–223. The energy-shortfall trace removes a nominal 0.040 J per-task allocation during tasks 96–255. Combined applies both. The synthetic reservoir begins at 22 J with a 2 J reserve.

All physical engine calls execute at 612 MHz. Actions at 306, 408, and 510 MHz replay frozen p95 latency and energy from the profile table while still invoking the selected adapted engine. The experiment is therefore compute-HIL: model choice and inference are physically exercised, but per-task frequency and the resource plant are not.

Table 2: Predeclared smoke cell. V1 was stopped for controller instability despite its score. V2 uses the separately frozen stability invariant.

Policy	Completed	Utility	Profiled J	Model switches
Best static	192/320	0.5175	10.780	0
Fixed schedule	192/320	0.5175	10.780	0
Adaptive DVFS	288/320	0.7775	17.030	0
Joint v1	320/320	0.8594	15.716	24
Joint v2	320/320	0.8525	14.343	2

Each scenario reuses the same 320 tiles. The matrix contains 76,800 task events, not 76,800 independent samples; 67,736 events invoke an engine after safety sheds. We predeclare a single smoke cell—combined trace, near workload, $c = 5$ —and require 320 joint-policy completions, at most four model transitions, zero TensorRT decision mismatches, zero profiled safety violations, and no answer key on the target before expanding the matrix.

4.4 Integrity and telemetry

All 240 runs complete with zero TensorRT decision mismatches against sealed references and zero profile-based safety violations. The maximum probability difference is 0.008114. The independent v2 smoke and its later matrix repetition match exactly for completion, utility, profiled energy, model/frequency residency, and switch counts for all policies. During the 501.5-second full matrix, active-GPU telemetry averages 7.481 W, peaks at 9.185 W and 52.812°C, and uses at most 2,078 MB system memory. These readings establish execution integrity, not spacecraft thermal behavior.

5 A Negative Result: Dwell Is Not Stability

The first frozen controller uses an eight-task minimum model dwell but no sustained service-rate release condition. In the smoke cell it appears strong on aggregate metrics: it completes 320 tasks with normalized utility 0.8594 and 15.716 profiled joules, versus 288 tasks, 0.7775 utility, and 17.030 joules for adaptive DVFS (Table 2). All engine parity and safety checks pass.

The event trace invalidates the controller. At medium/408, p95 service time is 7.582 ms while a near-load task arrives every 6 ms. The controller can finish an individual task before its immediate deadline, yet it cannot sustain the stream. Roughly three medium tasks consume accumulated queue slack; eight tiny tasks rebuild it; dwell expires; greedy quality ordering promotes medium again. This cycle produces 24 model transitions during the 128-task disturbance (Figure 2).

We stop the planned v1 matrix. Expanding it would precisely measure a known control defect. The v1 protocol, source, scores, and append-only event logs remain immutable. Version 2 is designed from the label-blind action sequence, arrivals, deadlines, profile service times, and resource state—not tar-

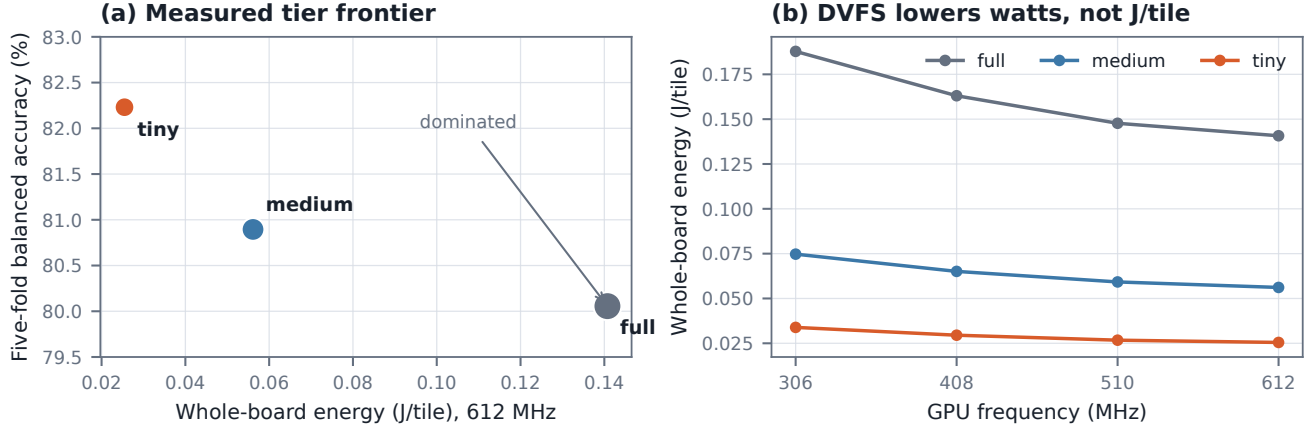


Figure 1: The full tier is dominated by tiny on the declared objectives. Medium and tiny remain non-dominated because medium has higher cloud recall while tiny has lower false-cloud rate, latency, and energy. Across all three tiers, 612 MHz draws more watts but uses the fewest whole-board joules per tile. Each model-frequency cell has one 30-second screening repetition.

Table 3: Aggregate over 60 repeated scenario cells per policy. “Completed” is out of 19,200 cell-tasks.

Policy	Completed	Mean utility	Profiled J	Model switches
Best static	15,360	0.679854	516.897	0
Fixed schedule	15,360	0.679854	516.897	0
Adaptive DVFS	17,856	0.795531	667.156	0
Joint v2	19,160	0.847458	646.405	8

get labels or scored mission utility—then frozen as a separate protocol.

6 Queue-Stable Controller Results

6.1 Regression smoke

The label-blind simulator predicts a demotion at task 96 and promotion at task 231. Real-engine execution reproduces that sequence. Joint v2 completes 320/320 tasks, reaches 0.8525 utility, uses 14.343 profiled joules, and makes two model transitions. Relative to adaptive DVFS, it completes 32 additional tasks, gains 0.075 utility, and uses 2.687 fewer profiled joules. All five predeclared expansion gates pass.

V2 utility is slightly below v1 because the stable controller holds tiny for 135 rather than 96 tasks. The transition reduction is not presented as a free accuracy gain; it deliberately trades some expected quality for a stable service regime.

6.2 Full sensitivity matrix

Table 3 sums each policy across repeated counterfactual cells. The totals are useful for consistency checks but are not fleet forecasts: cost ratios are arbitrary sensitivities and every cell reuses the same tasks.

Against adaptive DVFS, joint selection wins on utility in 12 cells, ties in 48, and loses in none (Figure 3). The boundary is sharper than the aggregate:

- At $c \in \{0.5, 1, 2\}$, the expected-loss rule fixes adaptive DVFS to tiny. The policies tie in all 36 cells.
- Every underloaded cell ties. When the preferred fixed model can sustain arrivals, model switching adds no value.
- At $c \in \{5, 10\}$, joint selection wins six of 12 cells per cost and ties the rest. Every win occurs at near load or overload.
- Under near load, gains require the 9.1 W power-cap transient: joint selection completes 32 additional tasks, adds 0.072–0.075 utility, and saves 2.687 profiled joules by using tiny/408 where medium/408 cannot sustain arrivals.
- Under overload on the benign and non-binding energy traces, it completes 106 additional tasks, adds 0.231–0.237 utility, and saves 3.864 profiled joules.
- Under overload with the power cap, it completes 188 additional tasks and adds 0.471–0.473 utility while using 1.363 more profiled joules. More useful work is not the same objective as less total energy.

Across 60 DVFS comparisons, joint selection uses less profiled energy in eight, more in four, and the same in 48. It never exceeds two model transitions in a cell: four cells contain two, and 56 contain none. It completes all 320 tasks in 50 cells and 316 in ten overloaded power-cap cells.

6.3 Null controls

The energy-shortfall trace exactly matches benign in completion, utility, energy, model residency, and frequency residency. Its 22 J starting reservoir makes the disturbance non-binding; it is a null control, not evidence of battery management.

Best-static and fixed-schedule are also identical in every cell. The nominal traces contain no predictable time-varying regime, so the precomputed schedule has nothing to exploit. This experiment compares reactions to unannounced disturbances and workload pressure. It does not test whether live

Negative result to invariant: v1 chatters; v2 demotes once and releases once

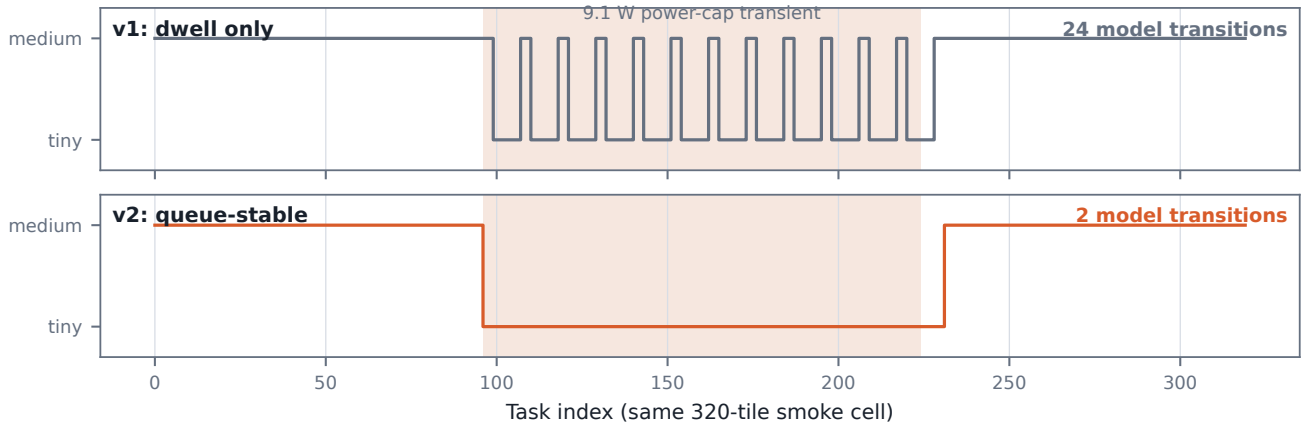


Figure 2: Model residency in the same combined/near/c = 5 smoke cell. An eight-task dwell suppresses per-task chatter but does not create a stable release condition (v1). The v2 service-rate filter demotes at the disturbance and promotes only after sustained stability and slack evidence.

adaptation beats a strong orbit-aware schedule built from a reliable recurring forecast.

7 What the Result Means

7.1 A software lever distinct from DVFS

This experiment gives a concrete answer to “why not just throttle the clock?” The measured profile shows that 612 MHz minimizes joules per tile for every tier. A lower frequency is useful only when a peak-power constraint rules out the fastest action; it then stretches service time. Under the 9.1 W transient, medium/408 is power-feasible but not service-stable at near load. Tiny/408 changes the graph, meets the arrival rate, and retains useful classification behavior. Clock control alone cannot reach that operating point because its model is fixed.

The solution is software-only in the limited systems sense that selection, hysteresis, admission, and engine dispatch need no new accelerator. It is not hardware-independent: it relies on precompiled target-specific engines, measurable rails or proxy state, resident memory, and a hardware interface that can enforce or report the relevant limits.

7.2 When a model ladder is worth carrying

The matrix suggests five necessary commercial and engineering conditions:

1. At least two deployable modes lie on a measured accuracy/resource frontier.
2. Binding intervals exist in which the preferred model cannot meet load or a resource cap while a cheaper mode retains value.
3. Missed or falsely cleared events cost materially more than false alarms.

4. The constrained interval is valuable enough to exceed storage, integration, verification, and switching complexity.
5. A live-state controller improves on the operator’s existing forecast schedule, power modes, and fault-management rules.

Thirty stored versions are not supported by this result. The screen retained two modes and rejected the full tier as dominated. Each additional mode expands qualification, artifact signing, rollback, and worst-case timing work. The practical question is the size of a small Pareto set, not the size of a model zoo.

The pattern can occur in terrestrial robots, drones, cameras, and intermittently powered edge systems, but actuator energy and charging access may dominate their budgets. Those markets require their own traces and loss functions; spacecraft rhetoric is not transferable evidence.

8 Limitations and Next Gates

This report intentionally draws a line around what was measured.

- **Synthetic plant and traces.** No operator bus telemetry, calibrated spacecraft battery, enclosure thermal model, attitude/eclipse history, or FDIR interaction is present.
- **Profile-replayed frequency.** Every physical inference runs at 612 MHz. Lower-frequency latency and energy are replayed from one profile run per model-frequency cell; real per-task DVFS transition overhead is absent.
- **Repeated data.** All 60 scenarios reuse one 320-tile shard. Win counts describe deterministic matrix coverage and do not provide confidence intervals or statistical significance.
- **Quality leakage at population level.** Aggregate priors and the scoring shard come from the same locked

Nomos minus adaptive-DVFS normalized utility across all 60 cells

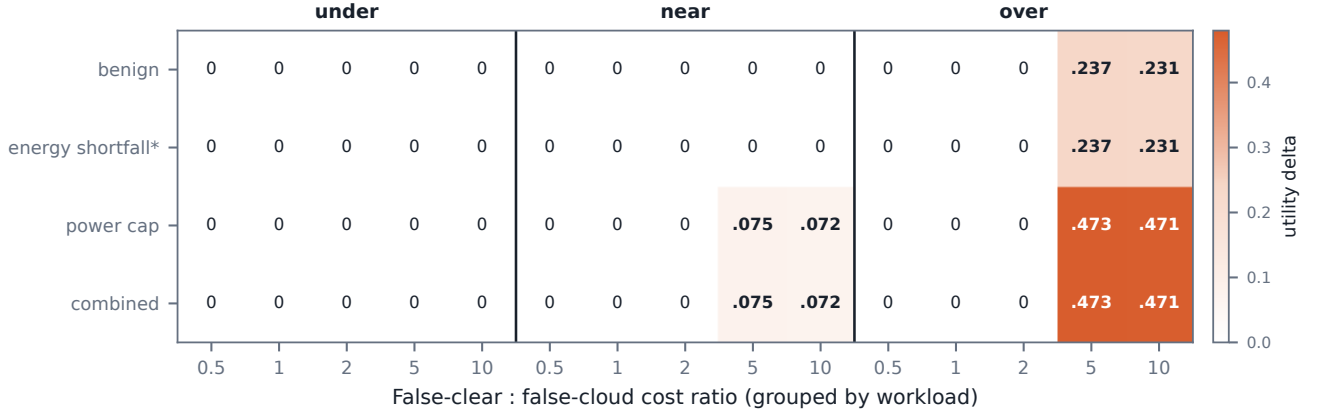


Figure 3: Normalized mission-utility difference between joint model/DVFS selection and adaptive DVFS. Zeroes are substantive nulls, not missing values. Gains appear only at high false-clear costs under near or over load. The frozen energy-shortfall trace is non-binding and exactly duplicates benign outcomes.

16-acquisition target population. The controller sees no labels, but a confirmatory study requires unseen acquisitions.

- **Measurement quality.** VDD_IN is onboard rail telemetry rather than an external calibrated meter. Source- and adapted-weight graphs are shape-identical, but the frequency sweep uses source weights.
- **Weak schedule test.** The fixed schedule equals static because the nominal trace has no recurring regime. A forecast-aware schedule remains untested.
- **No flight assurance.** Radiation effects, engine corruption, watchdog behavior, worst-case execution time, signed update/rollback, and integration with flight safety logic are outside scope.

The next decisive experiment needs an operator trace and loss model before more controller sophistication. Replay a time-aligned payload-demand, rail, temperature, attitude, and power-mode trace; run real clock transitions; meter the board externally; reserve unseen acquisitions; compare against the operator’s actual fixed schedule and governor; and predeclare the mission-level acceptance rule. A thermal chamber can then test a fitted plant. Only after those gates would flatsat or flight-language claims be warranted.

9 Reproducibility Statement

The publication page provides the paper source, figure generator, frozen v1/v2 protocols, tier/profile summaries, scored smoke summaries, full matrix summary, matrix aggregation, and the two smoke event traces. It also provides SHA-256 checksums. Key frozen hashes include:

Artifact	SHA-256
v1 protocol	64fd4992e640d92a71e6da57c6ed1c340fa797a7d1fde5775a4fd7302d53ff9f
v2 protocol	7f982362fe0635c627e6afe8c3fff83ad41385d354b3253b6bd42a522dfe67c
v2 controller	22653dc778647311e9ab96c63e2adf7c69c4149f30d9338ad8eef66226c668ba
full matrix	2a7de652bc9eeb703ce6677501c6c75a9a1ea2b634c0599ee5baf9cbbcb572a
aggregation	0dca8a278e01cbf46ee791e892d3ea337ff27dface47a229c00d3a2f2ff78a3b

Raw imagery is not redistributed; the acquisition identifiers and upstream dataset link are documented in the capsule. Serialized TensorRT engines are also omitted because they are tied to the target TensorRT/CUDA stack. The public artifact supports result inspection and figure regeneration, not a one-command reproduction of hardware execution.

10 Conclusion

DVFS and model selection are complementary controls. On this measured two-tier frontier, the joint controller adds utility only when the fixed preferred tier cannot sustain the workload or a peak-power cap and missed work is expensive. A naive dwell-based controller can look good in aggregate while oscillating; the failure trace matters more than its score. A queue-stability filter removes that limit cycle in the frozen smoke and exposes a narrow 12-win, 48-tie boundary against application-aware DVFS.

That boundary is the current result. It justifies a trace-driven operator validation, not a spacecraft claim. The most valuable next datum is not another synthetic win: it is whether a real operator has binding intervals, a non-dominated model ladder, and a loss function under which preserving inference through those intervals is worth the verification burden.

References

- [1] Gianluca Giuffrida, Lorenzo Diana, Francesco de Gioia, Gionata Benelli, Gabriele Meoni, Massimiliano Donati, and Luca Fanucci. CloudScout: A deep neural network for on-board cloud detection on hyperspectral images. *Remote Sensing*, 12(14):2205, 2020.
- [2] Andrew Du, Anh-Dzung Doan, Yee Wei Law, and Tat-Jun Chin. Domain adaptation for satellite-borne multispectral cloud detection. *Remote Sensing*, 16(18):3469, 2024.
- [3] M. Jehanzeb Mirza, Jakub Micorek, Horst Possegger, and Horst Bischof. The norm must go on: Dynamic unsupervised domain adaptation by normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14765–14775, 2022.
- [4] Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. Slimmable neural networks. In *International Conference on Learning Representations*, 2019.
- [5] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. In *International Conference on Learning Representations*, 2020.
- [6] Surat Teerapittayanon, Bradley McDanel, and H. T. Kung. BranchyNet: Fast inference via early exiting from deep neural networks. *arXiv preprint arXiv:1709.01686*, 2017.
- [7] Ziyang Zhang, Yang Zhao, Ming-Ching Chang, Changyao Lin, and Jie Liu. E4: Energy-efficient DNN inference for edge video analytics via early-exit and DVFS. *arXiv preprint arXiv:2503.04865*, 2025.
- [8] C. L. Liu and James W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM*, 20(1):46–61, 1973.
- [9] NVIDIA Corporation. Jetson orin nano series: Platform power and performance. Jetson Linux Developer Guide, release 36.2, 2024. Accessed 2026-08-24.