Decision Support for Resilient Foundation Model Scaling in Orbital Computing Systems

Michael Chung
School of Data Science
University of Virginia
hcz6jh@virginia.edu

Peter A. Beling
School of Data Science
University of Virginia
beling@virginia.edu

Tyler Cody
School of Data Science
University of Virginia
tcody@virginia.edu

***Abstract*—Future geospatial foundation-model pipelines will be shaped by the allocation of orbital infrastructure between data return, onboard computation, and the associated potential of orbital edge computing to serve those pipelines. These constraints induce distinct regimes for training, updating, and deploying foundation models, ranging from Earth-trained models operating on downlinked data to partially or fully in-space training enabled by orbital computing infrastructure. This paper develops a decision-support model that converts satellite launch growth into scenario-based estimates of downlink-oriented and compute-oriented capacity. Using payload launch data as an empirical launch-growth backbone, we compare logistic, exponential, and linear capacity forecasts under three compute-allocation assumptions. We then evaluate when cumulative in-space compute becomes order-equivalent to selected terrestrial foundation-model training thresholds. The results are not forecasts of specific in-space training programs. They provide an architectural planning tool for identifying when onboard computation becomes relevant to resilient geospatial foundation-model design.**

***Index Terms*—geospatial foundation models, orbital edge computing, resilience, satellite systems, forecasting**

I. Introduction

Geospatial foundation models increase the computational and data-management demands placed on remote-sensing pipelines [1], [2], [3]. At the same time, orbital edge computing is changing the architectural role of satellites [4], [5]. Space systems are increasingly considered not only as distributed sensors, but also as platforms for onboard processing, filtering, fusion, adaptation, and eventual machine-learning workloads.

This creates a systems-design question: should future geospatial foundation-model pipelines remain downlink-centered, or should they allocate increasing orbital capacity to onboard computation? The question should not be framed as a binary choice between training on Earth and training in space. It is a resilient build-out problem. Launch growth creates marginal orbital capacity that can be allocated across sensing, storage, communication, and computation.

Resilience here is a property of the development program over time, not only of a deployed pipeline. As geospatial foundation models scale, continued data collection, retraining, validation, and fielding depend on the evolving balance between downlink and orbital compute. Forecasting those resource trajectories is therefore necessary for resilient program design: it indicates when terrestrial training remains viable, when a hybrid architecture is warranted, and when shifting computation into orbit becomes necessary to sustain model development at scale.

This paper contributes a compact scenario framework for this planning problem. Historical launched-payload growth is used as the empirical backbone. Forward launch-capacity scenarios are translated into downlink-oriented and compute-oriented capacity under explicit allocation assumptions. Cumulative in-space compute is then compared with terrestrial foundation-model training thresholds. These thresholds are used only as compute-equivalent reference points, not as claims that geospatial foundation models require the same data, architecture, or sample complexity as language models.

II. Analytic Framework

Let S_t denote launched satellite payload capacity in year t. In this implementation, S_t is measured as annual launched payload count from the GCAT payload catalog. It is a proxy for orbital infrastructure growth, not a direct measurement of available compute.

For a compute allocation fraction $\alpha_t \in [0, 1]$, the annual compute-oriented and downlink-oriented satellite capacity proxies are

$$S_t^C = \alpha_t S_t, \tag{1}$$

$$S_t^D = (1 - \alpha_t) S_t. \tag{2}$$

Effective annual in-space training compute is modeled as

$$C_t = S_t \alpha_t \eta_t H_t \rho \tau, \tag{3}$$

where η_t is per-satellite peak compute throughput in FLOPs/s, H_t is the operating duty factor, ρ is the sustained-utilization factor, and τ is seconds per year. The cumulative effective training budget is

$$\widetilde{C}_e(T) = \sum_{t=t_0}^{T} C_t. \tag{4}$$

Effective annual downlink capacity is modeled as

$$D_t = S_t (1 - \alpha_t) \beta_t H_t \tau, \tag{5}$$

where β_t is per-satellite downlink throughput. The model is deliberately simple. Its purpose is to expose architectural sensitivity to launch growth and allocation assumptions. Table I

lists the fixed scenario parameters used in the illustrative calculations. The compute-throughput assumption is intentionally conservative and is meant to be representative of embedded AI hardware classes rather than terrestrial datacenter accelerators [6].

III. Data and Forecast Construction

The empirical input is the General Catalog of Artificial Space Objects (GCAT) payload catalog [7], a meta-database of publicly available launch data, filtered to successful payload launches with launch dates through 2025 [7]. Annual payload counts define S_t. Payloads are also grouped into broad categories for visualization. The category split is descriptive only. It should not be interpreted as measured onboard training capacity.

Figure 1 shows the annual launched-payload history from 1990 through 2025. The sharp post-2020 increase is dominated by communications payloads. This matters because the near-term growth of orbital infrastructure is currently communication-heavy, even if future architectures allocate some marginal capacity to compute.

Three forward curves are fit to recent annual payload counts: logistic, exponential, and linear. The logistic curve represents saturation of annual launch capacity, the exponential curve represents continued compounding growth, and the linear curve represents continued additive growth. These are scenario envelopes, not predictions.

The fitted scenarios are then combined with three values of α: downlink-dominant (1%), balanced (10%), and compute-dominant (50%). These three allocation settings are used throughout the remainder of the paper to separate architectural choice from launch-growth uncertainty. Figure 3 uses the logistic forecast to isolate the effect of allocation on the same underlying launched-capacity trajectory.

IV. Compute-Equivalent Reference Models

The reference thresholds are anchored to published dense-transformer training runs from GPT-3, LLaMA 2, and LLaMA 3 [8], [9], [10]. These model families provide recognizable milestones across early large-scale foundation models, widely used open-weight baselines, and more recent high-parameter systems. They are used here only as standardized compute markers.

For a model with parameter count N and training tokens D, the standard dense-transformer training-compute approximation is

$$F(N, D) = 6ND. \tag{6}$$

Following the accounting conventions used in those model reports [8], [9], [10], this gives a set of recognizable compute-equivalent thresholds. These thresholds are not geospatial training requirements. They are order-of-magnitude reference points for interpreting cumulative orbital compute. Table II lists the reference models used in this paper.

For forecast family k and allocation α, define the crossover year

$$T_k^*(N, D, \alpha) = \min\{T : \widetilde{C}_{e,k,\alpha}(T) \geq 6ND\}. \tag{7}$$

TABLE I
Scenario Parameters

Symbol	Value	Interpretation
η_t	150 TFLOPs	per-satellite compute throughput
β_t	90 Gbps	per-satellite downlink throughput
H_t	0.5	effective annual duty factor
ρ	0.10	sustained-utilization factor
t_0	2025	compute accumulation start year
α	0.01, 0.10, 0.50	allocation scenarios

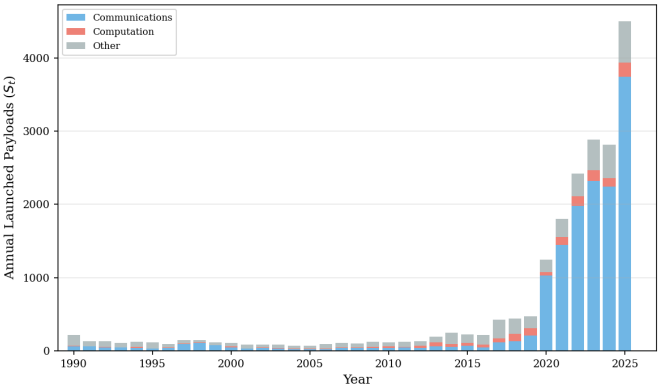

Fig. 1. Annual launched payload capacity by broad category. The figure shows the empirical launch-growth backbone used to construct S_t. Category labels are coarse and are used only to describe historical payload composition.

V. Illustrative Results

Figure 4 and Table III should be read together. Figure 4 shows the cumulative-compute trajectories relative to the benchmark thresholds, while Table III reports the first crossover year for each forecast-allocation pair. The result is best interpreted by threshold bands. Small-model reference thresholds are crossed early under most scenarios. Larger-model thresholds remain sensitive to both the launch-growth forecast and the compute-allocation fraction.

Table III reports crossover years. Under the balanced $\alpha = 10\%$ assumption, the GPT-3-scale threshold is reached around 2027–2028 across the three launch-growth forecasts. The LLaMA 3 70B threshold is reached only in the mid-2030s under exponential growth and around the 2060s under logistic or linear growth. The LLaMA 3 405B threshold is reached by 2043 only under the exponential forecast at $\alpha = 10\%$; otherwise it is outside the 2070 horizon except under compute-centric assumptions.

The main planning implication is that the first relevant transition is not full in-space training of the largest frontier models. It is the earlier point at which onboard compute becomes large enough to support partial training, adaptation, filtering, representation updates, and selective retraining within geospatial pipelines. These intermediate workloads are more relevant to resilient architecture design than a single all-or-nothing question about complete model pretraining in orbit.

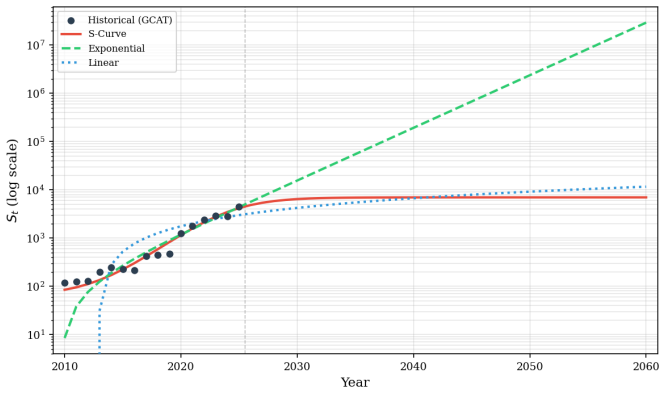

Fig. 2. Log-scale launch-capacity forecasts fit to recent GCAT payload counts. The vertical line marks the transition from historical data to forward scenario analysis.

TABLE II
COMPUTE-EQUIVALENT REFERENCE THRESHOLDS

Model	N	D	$6ND$ FLOPs
LLaMA 2 7B	7×10^{9}	2×10^{12}	8.40×10^{22}
GPT-3 175B	1.75×10^{11}	3×10^{11}	3.15×10^{23}
LLaMA 3 8B	8×10^{9}	1.5×10^{13}	7.20×10^{23}
LLaMA 2 70B	7×10^{10}	2×10^{12}	8.40×10^{23}
LLaMA 3 70B	7×10^{10}	1.5×10^{13}	6.30×10^{24}
LLaMA 3 405B	4.05×10^{11}	1.5×10^{13}	3.65×10^{25}

VI. DISCUSSION

The framework separates three questions that are often conflated. First, launch growth determines the scale of available orbital infrastructure. Second, architectural allocation determines how much of that infrastructure is compute-oriented rather than downlink-oriented. Third, workload design determines whether available compute is used for inference, adaptation, compression, representation learning, or full training. This decomposition is consistent with the broader orbital-edge-computing view that space systems are evolving from pure data-collection assets toward distributed computational infrastructure [4], [5].

This separation is useful for resilient geospatial foundation-model planning. A downlink-centered design may remain efficient when links, ground stations, and terrestrial compute are reliable. A hybrid design may be preferable when raw data return is intermittent, contested, expensive, or latency-constrained. A compute-centric design becomes relevant when onboard processing can preserve mission value even when the ground segment is degraded.

The results also show why architecture planning should begin before full in-space pretraining is practical. Once orbital compute is sufficient for meaningful adaptation or selective retraining, system designers must decide where to place data curation, model updating, fault recovery, and validation functions. These decisions affect verification, autonomy, cyber resilience, and lifecycle revalidation.

VII. LIMITATIONS

The model has five main limitations. First, payload count is only a proxy for orbital capacity. It does not encode mass, power, orbit, payload class, radiation tolerance, or mission architecture. Second, the mapping from payload growth to compute and downlink capacity is scenario-based. It is not directly observed. Third, the compute thresholds are language-model reference points, not geospatial foundation-model requirements. Fourth, the framework does not model inter-satellite networking, synchronization, memory capacity, thermal constraints, failure rates, or orbital dynamics. Fifth, the analysis treats η_t, β_t, H_t, and ρ as fixed, even though they will vary by platform and over time.

These limitations define the next modeling layer. The present framework should be used for early architectural reasoning, sensitivity analysis, and resilience-oriented planning. It should not be used as an operational forecast.

VIII. CONCLUSION

Resilient geospatial foundation-model architectures should be planned against the evolving trade-off between downlink capacity and onboard compute capacity. Historical launch growth provides an empirical basis for estimating orbital infrastructure expansion. Scenario-based allocation converts that expansion into architectural alternatives. Compute-equivalent thresholds then provide interpretable milestones for judging when in-space computation becomes relevant to foundation-model pipeline design.

The result is a decision-support framing rather than a prediction. It identifies when orbital compute should enter the design space for resilient geospatial foundation models and clarifies which assumptions drive that transition.

REFERENCES

[1] G. Mai, C. Cundy, K. Choi, Y. Hu, N. Lao, and S. Ermon, "Towards a foundation model for geospatial artificial intelligence (vision paper)," in *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, 2022, pp. 1–4.

[2] G. Mai, W. Huang, J. Sun, S. Song, D. Mishra, N. Liu, S. Gao, T. Liu, G. Cong, Y. Hu *et al.*, "On the opportunities and challenges of foundation models for geospatial artificial intelligence," *arXiv preprint arXiv:2304.06798*, 2023.

[3] M. Mendieta, B. Han, X. Shi, Y. Zhu, and C. Chen, "Towards geospatial foundation models via continual pretraining," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 16 806–16 816.

[4] B. Denby and B. Lucia, "Orbital edge computing: Nanosatellite constellations as a new class of computer system," in *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, 2020, pp. 939–954.

[5] Y. Zengshan, W. Changhao, G. Chongbin, L. Yuanchun, X. Mengwei, G. Weiwei, and C. Chuanxiu, "A comprehensive survey of orbital edge computing: Systems, applications, and algorithms," *Chinese Journal of Aeronautics*, vol. 38, no. 7, p. 103316, 2025.

[6] NVIDIA, "Jetson AGX Orin Series," https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/, 2024.

[7] J. C. McDowell, *General Catalog of Artificial Space Objects*, Jonathan's Space Report, 2026. [Online]. Available: https://planet4589.org/space/gcat/

[8] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

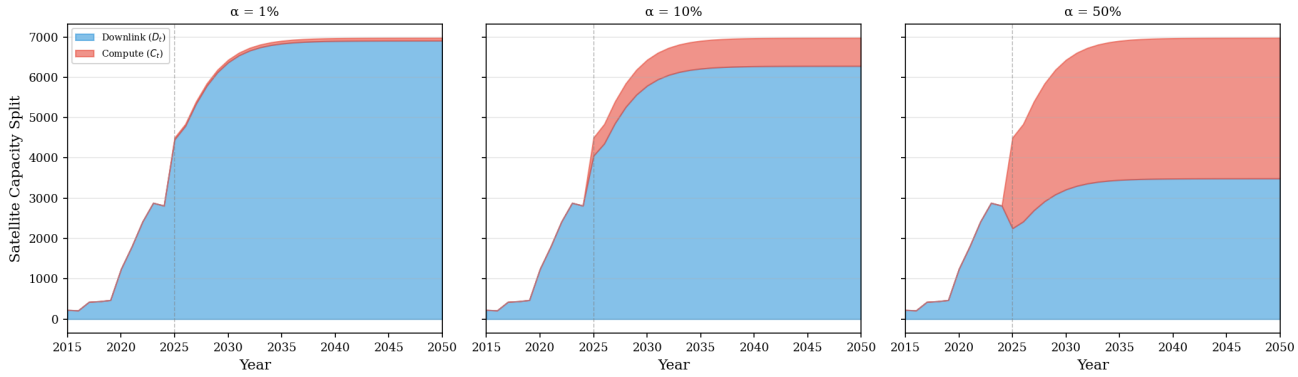

Fig. 3. Architectural resource split under the logistic launch-growth scenario. Blue indicates downlink-oriented capacity, and red indicates compute-oriented capacity. The panels isolate the effect of α on the allocation of the same underlying launched capacity.

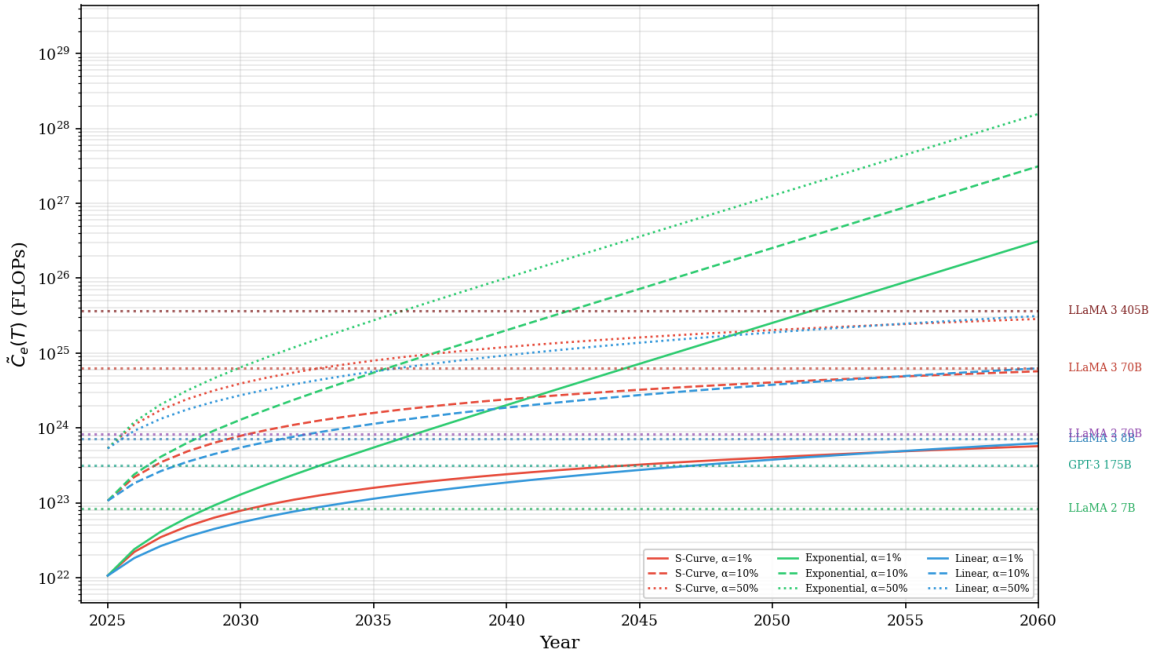

Fig. 4. Cumulative effective in-space compute versus terrestrial foundation-model training thresholds. Curves vary by launch-growth forecast and compute-allocation fraction α. Horizontal lines are compute-equivalent reference thresholds, not claims of geospatial model requirements.

[9] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[10] A. Dubey, A. Chowdhury, A. Bhattacharyya, A. Chowdhury, A. Arfeen, B. Chawla, B. Ganguly, B. Gupta, D. Sharma *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

TABLE III
CROSSOVER YEAR $T^*(N, D, \alpha)$ BY FORECAST AND COMPUTE ALLOCATION

Forecast	α	LLaMA 2 7B	GPT-3 175B	LLaMA 3 8B	LLaMA 2 70B	LLaMA 3 70B	LLaMA 3 405B
S-Curve	1%	2031	2045	2069	> 2070	> 2070	> 2070
S-Curve	10%	2025	2027	2030	2031	2064	> 2070
S-Curve	50%	2025	2025	2026	2026	2033	2070
Exponential	1%	2029	2033	2037	2037	2045	2052
Exponential	10%	2025	2027	2029	2029	2036	2043
Exponential	50%	2025	2025	2026	2026	2030	2037
Linear	1%	2033	2048	2064	2068	> 2070	> 2070
Linear	10%	2025	2028	2032	2033	2061	> 2070
Linear	50%	2025	2025	2026	2026	2036	2064