Compressed Decision ASIC for Planetary Rovers: A Hybrid Tabular–Transformer Architecture for Ultra–Low-Power Autonomy

Ing. Jose Luis Minich*
*Tab-Core Labs / GroovinAds, Córdoba, Argentina
Email: ai@tab-core.com

Abstract—**Planetary rovers must make safety-critical decisions under severe resource constraints: limited power budgets (few watts), harsh radiation environments, and communication delays (minutes to tens of minutes) make continuous ground supervision impractical. Concurrently, state-of-the-art AI decision models increasingly rely on large Transformer-based architectures whose computational and memory footprints far exceed the capabilities of space-qualified processors.**

We present a *compressed decision application-specific integrated circuit (ASIC) architecture* for planetary rovers, built around a hybrid design that combines (i) a set of efficient tabular multi-layer perceptron (MLP) cores optimized for scalar predictions from numeric and categorical features, and (ii) a tiny Transformer-based context core encoding short state and event histories. Rather than deploying large models on-board, we adopt a teacher–student compression framework: high-capacity teacher models on Earth (e.g., large language models (LLMs), vision backbones, and physics simulators) generate supervision signals over extensive rover scenarios; on-board, a compact ensemble of tabular experts and a tiny Transformer student approximates the teachers' decisions within the mission's operational envelope.

The resulting ASIC, termed *MultiTab-Core-T*, supports parallel expert evaluation for tasks including trajectory safety scoring, energy cost estimation, and anomaly detection, within latency in the few-millisecond range (e.g., 2–5 ms) and core power on the order of 1–2 W with modest board-level overhead, compatible with rover power budgets. We describe the architecture, interconnect, and memory organization, and derive analytical performance and energy estimates for representative decision workloads. We emphasize that these are analytical projections, not measurements from fabricated silicon. We also outline a roadmap toward aerospace-grade implementation and discuss applicability to terrestrial high-throughput decision workloads.

I. Introduction

Planetary exploration missions increasingly rely on mobile robotic platforms to autonomously navigate, collect scientific data, and respond to unexpected environmental conditions. Future missions will operate in more challenging terrains under more dynamic profiles, demanding more capable on-board decision systems. However, rover platforms face stringent constraints on power ($\sim$10 W continuous compute budget), mass, and thermal envelope, and must withstand harsh radiation environments over multi-year missions. These constraints make deploying modern large-scale AI models directly on-board impractical.

Concurrently, state-of-the-art decision models in Earth-based applications (robotics, online advertising, recommendation systems, financial risk scoring) increasingly adopt Transformer-based architectures and large language models (LLMs). While achieving impressive performance, these models typically require GPUs or TPUs consuming tens to hundreds of watts and tens of gigabytes of memory—far exceeding the capabilities of space-qualified compute platforms. Mars-Earth communication latency (4–24 minutes round-trip) further precludes continuous human-in-the-loop control for time-critical safety decisions.

We observe that many rover decisions can be formulated as:

$$\mathbf{decision} = f(\mathbf{x}_{\mathbf{tab}}, \mathbf{c}), \tag{1}$$

where $\mathbf{x}_{\mathbf{tab}}$ is a vector of tabular features (terrain slope, roughness, joint temperatures, battery state-of-charge, solar irradiance, link quality), and $\mathbf{c}$ is a compact representation of recent context (state history, operational mode, symbolic annotations). A large general-purpose model is not required on-board; rather, a compact domain-specific decision module *aligned* with a larger "teacher" model's behavior, but implemented in radiation-tolerant, ultra-low-power hardware, suffices.

We propose a dedicated *Compressed Decision ASIC* architecture for planetary rovers, termed *MultiTab-Core-T*. The design combines:

- **A set of identical, heavily optimized tabular MLP cores (MultiTab fabric), specialized for dense feature vectors and scalar predictions (trajectory risk, energy cost, anomaly scores).**
- **A tiny Transformer-based context core (Tab-Core-T), processing short sequences of contextual signals (recent states, event logs, symbolic tags) to produce a low-dimensional embedding that conditions the tabular experts.**

Rather than implementing a full-scale Transformer on-chip, MultiTab-Core-T operates as the *student* in a teacher–student compression framework. On Earth, high-capacity teacher models—including LLMs, deep perception backbones, and physics simulators—evaluate extensive rover scenarios offline, generating training targets. The on-board ensemble of tabular experts and tiny Transformer context core is trained to approximate the teachers' decisions within the mission's operational envelope. At deployment, only this compressed, hardware-aligned student ensemble executes on the rover, enabling high-quality decisions under tight power and latency constraints.

We emphasize that MultiTab-Core-T is *not* competitive with GPU/TPU systems for general-purpose training or full LLM inference. Our target is narrow decision workloads where domain structure and operational constraints enable orders-of-magnitude higher specialization.

Contributions

The main contributions of this paper are:

- **We introduce MultiTab-Core-T, a hybrid ASIC architecture combining a tiny Transformer-based context engine with an array of tabular MLP expert cores, tailored to decision workloads mixing short contextual sequences and rich tabular features in planetary rover missions.**

- We propose a teacher–student compression framework in which large Earth-side models (LLMs, vision backbones, physics simulators) generate training targets, enabling a compact on-board student ensemble to approximate their decisions within a restricted operational envelope.
- We describe the architecture, interconnect, and memory organization of the MultiTab-Core-T fabric supporting expert-parallel inference, and derive *analytical* (not measured) performance and energy estimates for representative decision workloads, comparing against CPU/GPU baselines.
- We outline a roadmap toward aerospace-grade implementation, discussing reliability, radiation tolerance, model update mechanisms, and applicability to terrestrial high-throughput decision workloads.

II. Decision Workloads in Planetary Rovers

We characterize the types of decisions planetary rovers must make and derive requirements for an on-board decision ASIC.

A. Rover Decision Tasks

Typical rover decision tasks include:

- Local path assessment: Scoring candidate trajectories for safety (slip, sinkage, tipping risk), energy cost, time to goal, and scientific value. Recent approaches employ learned terrain classifiers and risk models [1], [2], [3].
- Energy-aware planning: Deciding whether to move, reorient, or wait based on battery state-of-charge, predicted solar irradiance, thermal constraints, and mission timelines.
- Health and anomaly monitoring: Detecting deviations in actuator currents, temperatures, vibration patterns, or sensor readings indicating faults [4], [5].
- Scientific opportunity prioritization: Assigning priorities to observation targets based on scientific models, operational constraints, and mission goals.

Many tasks can be expressed as scalar prediction problems:

$$y = g(\mathbf{x}_{\text{tab}}, \mathbf{c}), \tag{2}$$

where y is a risk score, energy estimate, or utility value.

B. Constraints and Requirements

Rover platforms impose stringent constraints:

- Power and thermal envelope: Continuous compute budgets are typically a few watts (< 10 W), with tight thermal margins [6].
- Radiation environment: Electronics must tolerate total ionizing dose (TID) and single-event effects (SEE), favoring mature process nodes ($\geq$130 nm) and conservative design margins [7], [8].
- Latency: Safety-critical decisions (obstacle avoidance, stability checks) require sub-100 ms response times.
- Longevity and reliability: Missions last multiple years, demanding robust designs with well-understood failure modes and upset recovery.
- Limited communication: Round-trip times of 8–48 minutes and limited downlink bandwidth preclude continuous human supervision.

These requirements motivate a specialized, domain-aligned ASIC delivering high inference throughput and low latency at minimal power, while leveraging Earth-based high-capacity models during training.

III. MultiTab-Core-T Architecture

In this section we present the proposed hardware architecture. Figure 1 illustrates the main components at a high level. We refer to the overall accelerator family as *Tab-Core*, and to the specific hybrid tabular–Transformer ASIC presented in this work as *MultiTab-Core-T*.

A. Overview

The MultiTab-Core-T ASIC consists of three main subsystems:

1) A Context Engine based on a tiny Transformer (Tab-Core-T).
2) A MultiTab fabric containing N identical tabular MLP expert cores.
3) A Control and Interconnect block that orchestrates data movement, scheduling and communication with the rover's main CPU.

B. Context Engine (Tab-Core-T)

The Context Engine implements a compact Transformer encoder tailored to short sequences and low-dimensional embeddings:

- Ingesting sequences of recent rover states, commands, mode flags, and event tokens.
- Applying learned embeddings and positional encodings.
- Running a small number of self-attention and feed-forward layers.
- Producing a fixed-size context embedding $\mathbf{e} \in \mathbb{R}^d$.

Tab-Core-T is constrained by area and power budgets. A typical configuration comprises:

- Sequence length $L \leq 32$ tokens.
- Embedding dimension $d \in [32, 128]$.
- One or two attention layers with 2–4 heads.
- Quantized weights and activations (8-bit integer).

Crucially, Tab-Core-T is *not* a general-purpose LLM; it produces embeddings maximally useful for conditioning downstream tabular experts on operational context.

C. Tabular MLP Expert Cores (MultiTab)

The MultiTab fabric contains N identical tabular expert cores ($N \in [4, 16]$ typical). Each core implements a small multi-layer perceptron:

- Input dimension D_{in} accommodating $\mathbf{x}_{\text{tab}}$ concatenated with context embedding $\mathbf{e}$.
- One or two hidden layers with configurable width (32–128 units).
- Scalar output (risk score, cost estimate, anomaly score).

Each expert core is a deeply pipelined matrix–vector engine with local weight storage in SRAM, optimized for low-latency single-sample inference. The cores support:

- Parameter loading at boot or during model updates.
- Evaluating different models on different cores (expert-parallel mode).
- Evaluating the same model on multiple cores for batched workloads (data-parallel mode, if required).

D. Control, Interconnect and Memory Organization

A lightweight control processor orchestrates:

- Scheduling context embedding computations in Tab-Core-T.
- Distributing $\mathbf{e}$ and feature vectors $\mathbf{x}_{\text{tab}}$ to expert cores.
- Collecting and aggregating expert outputs.
- Communication with the rover's main CPU via standard interfaces (e.g., SpaceWire, low-voltage differential signaling).

The interconnect between Tab-Core-T and the MultiTab fabric is optimized for low-latency broadcast of $\mathbf{e}$ (typically 32–128 values) and routing of feature vectors. Bandwidth demands are modest compared to general-purpose accelerators moving large activation tensors.

Model parameters for Tab-Core-T and expert cores reside in on-chip SRAM or tightly coupled off-chip memory (radiation-tolerant MRAM or DRAM) with error-correcting codes (ECC). The design supports loading new parameters during mission via verified update packets from Earth, with rollback mechanisms.

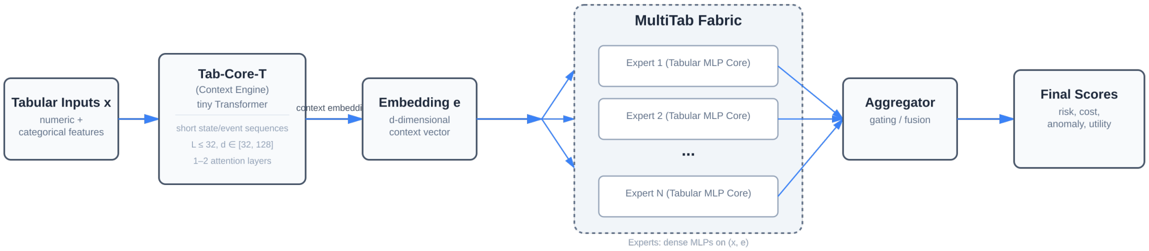


Fig. 1. High-level architecture of the MultiTab-Core-T stack. Tab-Core-T encodes recent context sequences into a context embedding $\mathbf{e}$, which is broadcast to a MultiTab fabric of expert MLP cores that consume tabular feature vectors $\mathbf{x}_{\text{tab}}$. An aggregator fuses their outputs into final scores (e.g., risk, cost, anomaly, utility).

IV. Teacher–Student Compression Framework

We outline how the proposed ASIC integrates into a teacher–student training pipeline executed on Earth prior to deployment and revisited during the mission.

A. Teacher Models and Scenario Generation

On Earth, we assume access to one or more high-capacity teacher models:

- Vision backbones coupled with physics-based terrain interaction simulators assessing trajectory risk and energy cost.
- Large language models integrating scientific goals, mission constraints, and contextual information into utility scores.
- Specialized models (anomaly detection, fault diagnosis) trained on extensive telemetry archives [4], [5].

Teachers evaluate large sets of simulated or replayed rover scenarios, generating supervision signals for the student ensemble. For each scenario, we construct:

- Tabular features $\mathbf{x}_{\text{tab}}$ from sensor readings, state estimators, and planning modules.
- Context sequences describing recent states, events, and annotations.
- Teacher outputs: trajectory risk scores, energy costs, anomaly labels, utility metrics.

B. Student Ensemble Training

The on-board student ensemble comprises:

- The tiny Transformer context core, parameterized by θ_T.
- A set of tabular experts, each parameterized by θ_i, $i = 1, \ldots, N$.

We train these components jointly or in stages to minimize discrepancies with teacher outputs. Loss functions include:

- Mean squared error (MSE) or cross-entropy between expert outputs and teacher scores.
- Distillation losses (e.g., Kullback–Leibler divergence) when teachers provide probabilistic outputs [9], [10].
- Regularization terms encouraging smoothness and robustness.

Expert specialization can be encouraged by routing scenario subsets to different cores or using techniques inspired by mixture-of-experts architectures [11], [12]. However, unlike runtime MoE routing, the MultiTab fabric can be configured statically at deployment, simplifying hardware design.

C. Model Updates During Mission

As data is collected and analyzed on Earth, teacher models can be refined and additional student training rounds performed. Updated parameters for Tab-Core-T and expert cores are uploaded to the rover, subject to:

- Integrity checks (cryptographic signatures, checksums).
- On-board validation using built-in test scenarios.
- Safe rollback to previous models if anomalies detected.

This decouples evolution of high-capacity Earth-side models from stable, resource-constrained on-board hardware.

V. On-Board Inference Flow and Integration

We now describe how the MultiTab-Core-T ASIC is integrated into the rover's control stack.

A. Sensor and Perception Front-End

Low-level perception (image processing, depth estimation, terrain classification) is handled by dedicated vision accelerators, FPGAs, or the rover's main CPU [2], [3]. These produce:

- Compact descriptors of candidate trajectories (slope, roughness, obstacles).
- Aggregated statistics over local terrain patches.
- Diagnostic features from actuators and sensors.

These descriptors are combined into tabular feature vectors $\mathbf{x}_{\text{tab}}$ and context tokens for the decision ASIC.

B. Decision Loop

A typical decision cycle:

1) The rover's CPU constructs a context sequence from latest states, events, and operational mode, sending it to Tab-Core-T.
2) Tab-Core-T computes embedding $\mathbf{e}$ and broadcasts it on the internal fabric.
3) For each candidate action (trajectory segment), the CPU sends feature vector $\mathbf{x}_{\text{tab}}$ to the MultiTab fabric.
4) Selected expert cores evaluate MLPs on $(\mathbf{x}_{\text{tab}}, \mathbf{e})$ in parallel, producing scalar scores.
5) Control logic aggregates scores and returns them to the CPU, which applies mission policy to select the final action.

Because internal computations are local and data transfers are modest, end-to-end decision latencies in the few-millisecond range (on the order of 2–5 ms for moderate numbers of candidate actions; see Section VI) are attainable.

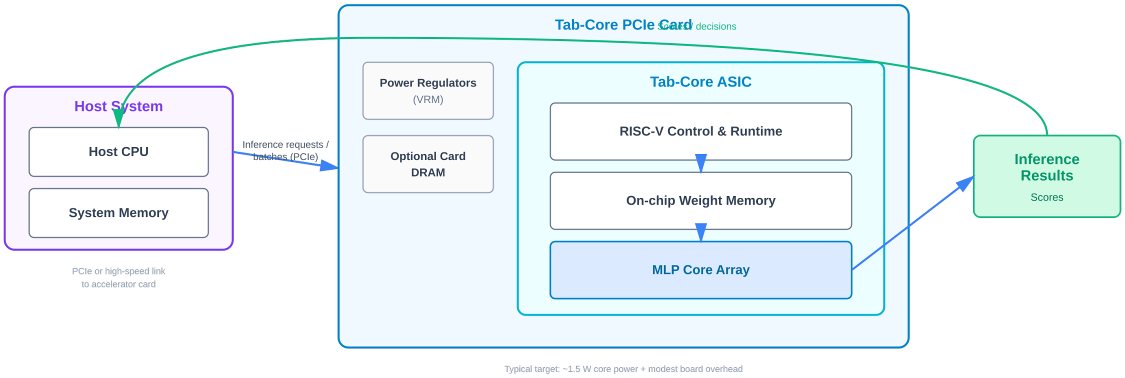


Fig. 2. System-level integration of the Tab-Core accelerator. The host CPU sends batched inference requests over PCIe (or a similar high-speed link) to the Tab-Core PCIe card. On the card, the Tab-Core ASIC—including RISC-V control, on-chip weight memory, and the MLP core array—executes the MLP workloads and returns scores back to the host. Typical target: ≈1–2 W core power plus modest board-level overhead (power regulators, optional DRAM).

C. Fail-Safe and Fallback Modes

The ASIC includes monitoring mechanisms (watchdog timers, output consistency checks, ECC error counters). If anomalies are detected (repeated timeouts, ECC faults, output divergence), the rover can:

- Enter conservative fallback mode using simpler CPU-based heuristics.
- Halt motion and request guidance from Earth.
- Attempt soft reset of the decision ASIC.

This ensures failures in the compressed decision module do not directly translate into unsafe rover behavior. The CPU retains ultimate authority over motion commands.

VI. Analytical Performance and Energy Model

Important disclaimer: The following analysis is based on register-transfer level (RTL) design analysis, power-performance-area (PPA) estimates from synthesis tools, and system-level modeling. As of December 2025, we have *not* fabricated silicon. All performance and energy numbers are analytical projections derived from our design implementation and conservative assumptions about the target process node. These are *not* measurements from manufactured chips. Real silicon characterization in 2026 will provide actual numbers, which we commit to publishing openly even if they differ significantly from these targets. Values should be interpreted as order-of-magnitude projections with substantial uncertainty ($\pm 50\%$ or more).

A. Operation Counts

Consider a representative student configuration:

- Tab-Core-T: one attention layer, one feed-forward layer, $L = 16$ tokens, embedding dimension $d = 64$.
- Tabular experts: input dimension $D_{\text{in}} = 96$ (64 from embedding, 32 from tabular features), one hidden layer of width $H = 64$, scalar output.

Under standard Transformer and MLP formulations, we define:

- Transformer operations: $O_T \approx 2Ld^2$ multiply–accumulate (MAC) operations for attention and feed-forward layers (ignoring softmax and layer normalization overhead for simplicity).
- Expert MLP operations: $O_E \approx D_{\text{in}} \cdot H + H \cdot 1$ MACs per forward pass, where H is the hidden layer width.

For $L = 16$, $d = 64$, $D_{\text{in}} = 96$, $H = 64$:

$$O_T \approx 2 \cdot 16 \cdot 64^2 \approx 131{,}000 \text{ MACs},$$
$$O_E \approx 96 \cdot 64 + 64 \cdot 1 \approx 6{,}200 \text{ MACs}.$$

Total operations per decision: If N experts are evaluated per decision:

$$O_{\text{total}} \approx O_T + N \cdot O_E. \quad (3)$$

For $N = 4$: $O_{\text{total}} \approx 131{,}000 + 4 \times 6{,}200 \approx 156{,}000$ MACs.

B. Throughput and Latency

Assumptions: We assume a mature process node (130 nm or 180 nm radiation-tolerant technology) with conservative clock frequencies and deeply pipelined datapaths. We estimate an effective throughput:

$$P \approx 100 \text{ MMAC/s (million MACs per second) per core}, \quad (4)$$

accounting for pipeline stalls, control overhead, and quantization effects. We assume similar throughput for both Tab-Core-T and expert cores.

Latency derivation: From operation counts and throughput, we compute execution time per block as $t = O/P$:

$$t_T \approx \frac{O_T}{P} \approx \frac{131{,}000}{100 \times 10^6} \approx 1.3 \text{ ms} \quad \text{(context embedding)}, \quad (5)$$
$$t_E \approx \frac{O_E}{P} \approx \frac{6{,}200}{100 \times 10^6} \approx 62\ \mu\text{s} \quad \text{(per expert)}. \quad (6)$$

Key architectural factors enabling low latency:

1) Embedding reuse: The context embedding $\mathbf{e}$ is computed once and reused across all N experts.
2) Expert parallelism: All N experts execute concurrently on separate cores.
3) Dominant latency: The overall latency is $\max(t_T, t_E) \approx 1.3$ ms, not the sum.

Including additional interconnect delays, control overhead, and memory access latency, the total decision time is estimated at ~ 2–5 ms for $N = 4$ experts. This range is a conservative illustrative example; it remains well within rover safety-critical requirements (< 100 ms).

TABLE I
ILLUSTRATIVE ENERGY PER DECISION FOR A STUDENT MODEL WITH ONE TINY TRANSFORMER CONTEXT ENGINE AND $N = 4$ TABULAR EXPERTS. VALUES ARE *not measured*; THEY ARE ANALYTICAL ESTIMATES DERIVED FROM OPERATION COUNTS AND ASSUMED ENERGY VALUES, ASSUMING THE SAME STUDENT MODEL RUNNING ON DIFFERENT PLATFORMS.

Platform	Energy/decision	Relative (vs CPU)
Embedded CPU (baseline)	∼30 mJ	1.0×
Low-power GPU	∼10 mJ	3×
MultiTab-Core-T (est.)	∼2 mJ	15×

C. Energy per Decision

Energy model definition: We model energy per decision as the sum of computational energy and memory access energy:

$$E_{\text{dec}} \approx O_{\text{total}} \cdot E_{\text{MAC}} + B_{\text{mem}} \cdot E_{\text{mem}}, \tag{7}$$

where:

- E_{MAC} **is energy per MAC operation,**
- E_{mem} **is energy per bit of memory access,**
- B_{mem} **is total memory bits accessed (parameters and activations).**

Assumptions: For a mature radiation-tolerant process, we assume conservative energy values based on prior work [13]:

- $E_{\text{MAC}} \approx 10$ **pJ/MAC (8-bit integer MAC, typical for mature nodes ≥130 nm),**
- $E_{\text{mem}} \approx 5$ **pJ/bit for on-chip SRAM access with ECC.**

Example calculation: For the configuration above ($O_{\text{total}} \approx 156{,}000$ **MACs,** $B_{\text{mem}} \approx 50{,}000$ **bits):**

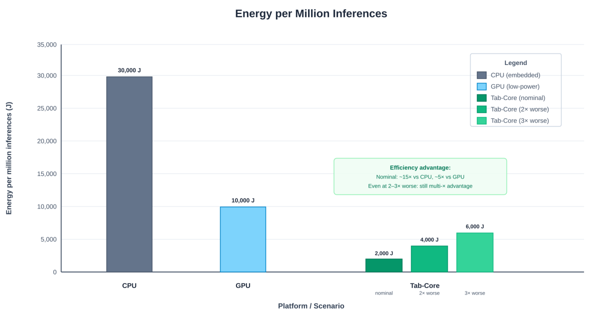
$$E_{\text{dec}} \approx 156{,}000 \times 10\ \textbf{pJ} + 50{,}000 \times 5\ \textbf{pJ} \approx 1.8\ \textbf{mJ} \approx 2\ \textbf{mJ}. \tag{8}$$

Rounding to ∼2 mJ per decision provides a conservative round number for comparison. Table I compares the hypothetical MultiTab-Core-T ASIC against CPU/GPU baselines running the *same student model*. All values are illustrative analytical estimates derived from operation counts and conservative energy-per-operation assumptions; specialization and data locality enable lower energy per decision.

These ratios (approximately $15\times$ **lower energy than the embedded CPU baseline and about** $5\times$ **lower than the low-power GPU for the same student model) are consistent with prior domain-specific accelerator studies [13], [14], [15]. We emphasize again that these are order-of-magnitude projections, not measurements from fabricated silicon. Detailed circuit-level modeling and silicon measurements are required for validation.**

D. Applicability of the Analytical Framework

The methodology presented here—defining a representative configuration, counting operations, assuming conservative throughput and energy parameters, and deriving latency and energy estimates—is broadly applicable to other ASIC designs targeting specialized workloads. The key steps are: (1) specify the architecture and operational parameters (L**,** d**,** H**,** N**, etc.); (2) derive operation counts (**O_T**,** O_E**,** O_{total}**) from the model structure; (3) assume a conservative MAC throughput** P **and energy costs (**E_{MAC}**,** E_{mem}**) appropriate to the target process node; (4) compute latency as** $t = O/P$ **and energy as** $E = O \cdot E_{\text{MAC}} + B_{\text{mem}} \cdot E_{\text{mem}}$**; and (5) compare against CPU/GPU baselines. This analytical framework enables early-stage design-space exploration and provides defensible order-of-magnitude projections even in the absence of fabricated silicon, provided all assumptions are explicitly stated and appropriate disclaimers are included.**



Analytical projections based on RTL + PPA estimates, not measured silicon.
Real silicon characterization expected Q4 2026.

Fig. 3. Projected energy per million inferences for an embedded CPU, a low-power GPU, and the Tab-Core ASIC. Under the illustrative configuration in Table I, Tab-Core targets around 15× lower energy per million inferences than the CPU baseline (2,000 J vs 30,000 J) and about 5× compared to a low-power GPU (2,000 J vs 10,000 J). Even under pessimistic assumptions (2–3× worse than nominal), Tab-Core would still retain a multi-× advantage. All figures are based on analytical RTL + PPA estimates, not measured silicon.

E. Limitations, Risks, and Uncertainty

Pre-silicon estimates carry inherent uncertainty. Our projections are based on RTL synthesis results, PPA estimates from commercial EDA tools (targeting a 130–180 nm radiation-tolerant process), and conservative assumptions about clock frequencies, memory access patterns, and control overhead. However, analytical models systematically underestimate certain effects:

- **Leakage power: Static power in deep-submicron processes can dominate at lower activity factors. Our estimates include leakage approximations, but real silicon may exhibit 1.5–2× higher leakage, especially at elevated temperatures.**
- **IO and board-level overhead: The chip-level energy estimates do not include off-chip memory interfaces, voltage regulators, clock distribution to the board, or cooling. A complete system will consume additional watts beyond the core.**
- **Process variation and aging: Corner-case analysis (slow-slow, fast-fast) and aging effects (NBTI, HCI) are included in our margin analysis, but real parts may exhibit wider spread than models predict.**
- **Radiation-induced upsets: Triple modular redundancy (TMR) and ECC add area and power overhead; our estimates include these, but actual rates of single-event upsets (SEU) in space may require more aggressive mitigation, increasing power further.**

Pessimistic scenario: If real silicon lands 2–3× worse than our nominal projections due to these factors, the efficiency advantage over CPU/GPU baselines would narrow but remain substantial (approximately 5**–**$7.5\times$ **vs CPU and** 1.7**–**$2.5\times$ **vs GPU, instead of the nominal** $15\times$ **and** $5\times$**). Even under such pessimistic assumptions, Tab-Core retains a multi-× advantage. We commit to publishing post-silicon measurements openly, regardless of outcome.**

Comparison fairness: All comparisons are at the chip or accelerator-card level, running the *same student model*. We do not compare a specialized ASIC against a general-purpose GPU running a full-scale teacher LLM; such comparisons would be misleading. Real deployed systems include additional system-level overhead (power supplies, cooling, networking) that affects all platforms and slightly reduces net efficiency gains.

Figure 3 illustrates the projected energy per million inferences for representative CPU and GPU baselines compared to Tab-Core under nominal and pessimistic assumptions.

VII. Scope and Limitations

This work focuses on architectural design and analytical characterization of MultiTab-Core-T. We do not present measurements from fabricated silicon. All performance and energy results are analytical estimates based on operation counts and assumed implementation parameters (process node, clock frequency, energy per MAC/memory access). These should be interpreted as order-of-magnitude projections, not validated measurements.

Our teacher–student framework targets *restricted operational envelopes*: the student ensemble approximates teacher decisions over terrains, environmental conditions, and mission profiles covered by simulation and archived telemetry. Extrapolation far beyond this envelope is not guaranteed and would require retraining or fallback to conservative heuristics.

We assume perception front-ends (stereo vision, depth estimation, terrain classification) provide compact descriptors to the decision ASIC. End-to-end perception (raw pixels to decisions) is out of scope. Similarly, we do not claim general-purpose LLM capability or competitiveness with GPUs/TPUs for full-scale language modeling or vision tasks.

VIII. Aerospace-Grade Design Considerations

Transitioning from architectural proposal to flight-qualified implementation requires addressing:

- Radiation-tolerant process: Selection of appropriate libraries (e.g., IBM 130 nm SOI, TSMC 180 nm hardened) [7], [8].
- Error mitigation: Error-correcting codes (ECC) for memories; triple modular redundancy (TMR) for critical control logic; watchdog timers and scrubbing [8].
- Robust physical design: Conservative clocking, power distribution with margins for aging and temperature variation (-55°C to $+125^\circ$C).
- Verification and validation: Fault injection campaigns, radiation testing (proton, heavy ion), hardware-in-the-loop simulation, long-duration stress testing.

We envision a phased roadmap: early prototypes in commercial processes validate the architecture and training framework in terrestrial settings; subsequent migration to radiation-tolerant processes and integration into rover avionics test platforms; final flight qualification.

IX. Broader Applications

Although we focus on planetary rovers, MultiTab-Core-T applies to terrestrial workloads where:

- Decisions depend on rich tabular features plus short contextual sequences.
- Throughput and latency requirements are extreme (millions of decisions per second).
- Power and cost constraints make large general-purpose accelerators unattractive.

Examples include real-time ad bidding [16], large-scale recommender systems [17], financial risk scoring, and industrial anomaly detection. The same teacher–student compression framework applies, with datacenter-scale teachers and edge-deployed MultiTab-Core-T accelerators.

X. Related Work

AI for planetary rovers. Prior work explored machine learning for rover autonomy. Goldberg et al. [2] developed stereo vision for Mars rovers. Angelova et al. [3] applied learning-based terrain classification. Ono et al. [1] proposed risk-aware path planning using learned terrain models. Recent efforts integrate deep learning for visual navigation and hazard detection [18], [19]. These focus on algorithms and software pipelines on existing processors/FPGAs. We propose a dedicated ASIC architecture aligned with a teacher–student compression framework.

Edge and low-power accelerators. Domain-specific accelerators for deep learning include Google's TPU [15], Eyeriss [14], and various NPUs. Most target convolutional and Transformer workloads for vision or general inference. Horowitz [13] analyzed energy costs of arithmetic vs. memory access, motivating data locality. Our design targets decision workloads dominated by tabular MLPs plus a tiny Transformer, emphasizing extreme energy efficiency under rover constraints.

Knowledge distillation and compact models. Hinton et al. [9] introduced knowledge distillation for model compression. Sanh et al. [10] applied distillation to BERT, creating DistilBERT. Recent work distills large language models for edge devices [20]. Mixture-of-experts architectures [11], [12] enable specialization, though typically with dynamic routing. We combine distillation with hardware tailored to tabular+context decision workloads, using static expert allocation.

Space-qualified computing. Radiation-hardened processors and FPGAs have been deployed on rovers [6]. Barth [7] and Kastensmidt et al. [8] survey radiation effects and mitigation. Iverson and Hummel [4] and Doran et al. [5] address fault detection for space systems. Our work extends this domain with a specialized decision accelerator integrating recent ML advances.

XI. Conclusion

We presented MultiTab-Core-T, a hybrid tabular–Transformer ASIC architecture as a compressed decision module for planetary rovers. By combining a tiny Transformer context engine with an array of efficient tabular MLP experts, and leveraging a teacher–student compression framework, the design targets high-quality rover decisions under strict power and latency constraints. We described the architecture, integration into the rover control stack, and a roadmap toward aerospace-grade implementations.

Status as of December 2025: All performance and energy results are analytical projections derived from RTL implementation and PPA estimates, not measurements from manufactured chips. We have completed detailed synthesis targeting a 130–180 nm radiation-tolerant process and derived conservative energy estimates. First silicon tapeout is planned for Q2 2026, with characterization results expected by Q4 2026. We commit to publishing measured results openly, even if they differ from these projections.

The same architectural principles apply to terrestrial high-throughput decision workloads (online advertising, recommendation systems, financial risk assessment) where tabular features and short context dominate, and extreme efficiency is required. A commercial-process variant targeting these markets is under parallel development.

Immediate next steps include: commercial-process prototyping (Q1 2026), experimental validation of the teacher–student framework on rover simulation datasets, radiation testing of IP blocks, and preparation for tapeout in radiation-hardened processes.

References

[1] M. Ono, M. Heverly, C. Rothrock, E. Almeida, R. Gaines, C. Dono, and M. Downs, "Risk-aware planetary rover operation: Autonomous terrain classification and path planning," in *Proc. IEEE Aerospace Conference*, 2015, pp. 1–10.

[2] S. B. Goldberg, M. W. Maimone, and L. Matthies, "Stereo vision and rover navigation software for planetary exploration," in *Proc. IEEE Aerospace Conference*, vol. 5, 2002, pp. 5–2025–5–2036.

[3] A. Angelova, L. Matthies, D. Helmick, and P. Perona, "Learning and prediction of slip from visual information," *Journal of Field Robotics*, vol. 24, no. 3, pp. 205–231, 2007.

[4] D. L. Iverson and R. B. Hummel, "Autonomous reasoning for space systems," in *Proc. IEEE Aerospace Conference*, 2008, pp. 1–14.

[5] H. D. Doran, J. E. Saleh, and D. E. Hastings, "Assessing the value of autonomy for fractionated spacecraft," in *Proc. AIAA Infotech@Aerospace Conference*, 2009, related to long-term autonomy and fault management.

[6] J. Caruso, T. Pham, A. Kiely, and J. Carsten, "Mars 2020 mission overview," in *Proc. IEEE Aerospace Conference*, 2017, pp. 1–10.

[7] J. L. Barth, "Space, atmospheric, and terrestrial radiation environments," *IEEE Transactions on Nuclear Science*, vol. 44, no. 6, pp. 1953–1964, 1997.

[8] F. L. Kastensmidt, L. Carro, and R. Reis, *Fault-Tolerance Techniques for SRAM-Based FPGAs*, ser. Frontiers in Electronic Testing. Springer, 2006, vol. 32.

[9] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015, nIPS 2014 Deep Learning Workshop.

[10] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019, presented at NeurIPS 2019 EMC2 Workshop.

[11] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," in *Proc. International Conference on Learning Representations (ICLR)*, 2017. [Online]. Available: https://openreview.net/forum?id=B1ckMDqlg

[12] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *Journal of Machine Learning Research*, vol. 22, no. 120, pp. 1–39, 2021. [Online]. Available: http://jmlr.org/papers/v22/21-0998.html

[13] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC)*, 2014, pp. 10–14.

[14] Y.-H. Chen, J. Emer, and V. Sze, "Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks," in *Proc. ACM/IEEE International Symposium on Computer Architecture (ISCA)*, 2016, pp. 367–379.

[15] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, R. Boyle, P. luc Cantin, C. Chao, C. Clark, J. Coriell, M. Daley, M. Dau, J. Dean, B. Gelb, T. V. Ghaemmaghami, R. Gottipati, W. Gulland, R. Hagmann, C. R. Ho, D. Hogberg, J. Hu, R. Hundt, D. Hurt, J. Ibarz, A. Jaffey, A. Jaworski, A. Kaplan, H. Khaitan, D. Killebrew, A. Koch, N. Kumar, S. Lacy, J. Laudon, J. Law, D. Le, C. Leary, Z. Liu, K. Lucke, A. Lundin, G. MacKean, A. Maggiore, M. Mahony, K. Miller, R. Nagarajan, R. Narayanaswami, R. Ni, K. Nix, T. Norrie, M. Omernick, N. Penukonda, A. Phelps, J. Ross, M. Ross, A. Salek, E. Samadiani, C. Severn, G. Sizikov, M. Snelham, J. Souter, D. Steinberg, A. Swing, M. Tan, G. Thorson, B. Tian, H. Toma, E. Tuttle, V. Vasudevan, R. Walter, W. Wang, E. Wilcox, and D. H. Yoon, "In-datacenter performance analysis of a tensor processing unit," in *Proc. ACM/IEEE International Symposium on Computer Architecture (ISCA)*, 2017, pp. 1–12.

[16] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction," in *Proc. ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1059–1068.

[17] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for YouTube recommendations," in *Proc. ACM Conference on Recommender Systems*, 2016, pp. 191–198.

[18] B. Rothrock, R. Kennedy, C. Cunningham, B. Pajarillo, M. Dietrich, and P. Backes, "SPOC: Deep learning-based terrain classification for mars rover operations," in *Proc. AIAA SPACE Conference and Exposition*, 2016.

[19] C. Cunningham, I. Nesnas, and W. Whittaker, "Improving slip prediction on mars using thermal inertia measurements," *Autonomous Robots*, vol. 41, no. 4, pp. 911–927, 2017.

[20] S. Sun, Y. Cheng, Z. Gan, and J. Liu, "Patient knowledge distillation for BERT model compression," in *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019, pp. 4323–4332.