JOURNALS | Journal of Collective Sciences and Sustainability

Volume 2 Issue 3 (September 2026)
J. Collect. Sci. Sustain., 2026, 2(3), 26406
DOI : https://doi.org/10.64189/css.26406

Research Article

OPEN ACCESS

Energy-Constrained AI Inference and Legacy Hardware Optimization for Sustainable ICT Energy Footprint Management

Parnika Thakur*

Department of Computer Engineering, Vishwakarma Institute of Technology, Pune, Maharashtra, 411037, India

*Corresponding Author
Parnika Thakur
parnikathakur27@gmail.com

Received: 05 July 2026
Revised: 29 July 2026
Accepted: 04 August 2026
Published Online: 06 August 2026

Citation

P. Thakur, Energy-Constrained AI inference and legacy hardware optimization for sustainable ICT energy footprint management, *Journal of Collective Sciences and Sustainability*, 2026, 2(3), 26406, https://doi.org/10.64189/css.26406.

Abstract

The rapid growth of generative AI has disrupted the conventional energy trends of data centers, resulting in sudden energy spikes and accelerating hardware replacement cycles. Although the industry's standard approach involves instant updates for new accelerators, such an approach results in essential environmental oversight because of the failure to consider manufacturing emissions. In this paper, we propose an energy-aware, software-oriented architecture to limit the energy consumption of local infrastructure and prolong the lifetime of legacy enterprise hardware. With the help of request-based model cascades and an emergency load-based admission control module, we are able to decrease the complexity of the models and the accuracy of their inference at times of increased loads. The efficiency of the proposed system is demonstrated through trace-driven numerical experiments using the real-world Alibaba Cluster Trace. Moreover, life cycle optimization modeling shows that retaining legacy hardware yields lower net lifecycle carbon emissions than premature replacement does on grids with abundant renewables.

Keywords: Artificial Intelligence; Sustainable Computing; Carbon-Aware Computing; Data Center Energy Management; Legacy Hardware Optimization; Model Cascading; Admission Control; ICT Energy Footprint.

1. Introduction

Over the past decade, the information and communication technology (ICT) industry has been able to effectively minimize environmental impact through aggressive infrastructural consolidation. The number of cloud computing instances and amounts of data increased more than fivefold between 2010 and 2018, whereas the overall energy usage of data centers demonstrated a negligible dependence on the growth factor and increased by only 6%.[1] This stability was possible because of the relocation of the IT workload into the hyperscale data centers, which were optimized to achieve maximum performance through the server virtualization and facilities' power consumption. However, the rise of generative artificial intelligence and large language models (LLMs) represents a fundamental challenge to this sustainability equation.[2] This pattern mirrors the Jevons paradox: as algorithmic efficiency improved and the marginal cost of computation decreased, computing became more accessible, and aggregate demand expanded, ultimately exceeding any local efficiency gains. As a consequence of such workloads, enterprise data center operators are forced to follow a strategy of hyper-accelerated obsolescence of the infrastructure and retire functional pieces of hardware such as legacy enterprise graphics processing units (GPUs) for cutting-edge high-density AI accelerators. This represents a serious environmental issue since the ICT industry pays attention solely to the operational energy consumption and efficiency of the power grid without considering the colossal Scope 3 embodied carbon footprint of semiconductor fabrication.[3]

Existing efforts to address these pressures largely treat query-level efficiency and infrastructure-level power management as separate problems. Model cascading and dynamic routing frameworks[4-6] optimize for inference cost, latency, or accuracy trade-offs at the level of an individual query but do not couple these decisions to a hard physical power ceiling at the rack or facility level. Conversely, carbon- and power-aware frameworks operate at the opposite end of the stack. Storage-layer optimizations such as AdaCache mitigate cloud workload pressure through block-level data management rather than inference-time computation. Grid-facing dispatchers such as Radovanović et al. shift the timing or location of entire batch workloads in response to grid-level carbon intensity but do not intervene in real time on individual, sub-second inference requests. Neither class of approach dynamically links per-request semantic routing decisions to a hard infrastructure power constraint nor uses precision degradation (from 16-bit floating point, FP16, through 8-bit integer, INT8, to 4-bit integer, INT4) as a real-time localized safeguard against rack-level PDU failure. The novelty of the present work lies precisely in this codesign: a microlayer request cascade and a macro-layer admission control engine operate jointly and are both mathematically governed by a forward-looking, lifecycle carbon-payback model that is explicitly optimized for extending the operational lifespan of legacy enterprise hardware, rather than treating power management and hardware retirement as independent decisions. Retirement of functional hardware in exchange for newly manufactured accelerators creates carbon debt, which will take years to be compensated depending on the carbon intensity of the local energy grid. Therefore, to address this contradiction, this paper offers a software-based architecture design that minimizes the local infrastructure's energy consumption while extending the lifespan of the hardware deployed at data centers. The validation of the proposed architecture design via trace-driven numerical simulation aligned with modern compliance standards reveals that algorithmic intelligence is able to provide sustainability for the local infrastructure, as maintenance of legacy hardware results in a net-negative carbon lifecycle compared with the accelerated hardware replacement schedule.[7]

2. Methods/Experimental Details

2.1 System architecture and hardware telemetry

The baseline of the system is defined using a server rack containing NVIDIA V100 enterprise GPUs as the e legacy node versus a modern alternative in the form of high-density NVIDIA H100 Tensor Core accelerators. Power usage profiles will be derived on the basis of multitenant cluster deployment attributes and architectural carbon footprints.[7,8] The initial operational boundaries are set by the thermal design power (TDP) of the corresponding accelerator nodes, with legacy hardware operating at TDP_{legacy} = 300 W and modern hardware operating at TDP_{new} = 700 W. Rack power is constrained by the hard upper cap, P_{cap}, of the localized power distribution unit (PDU):

$$P_{rack}(t) \leq P_{cap} \quad (1)$$

where $P_{rack}(t)$ is the instantaneous power consumption of the cluster at time t.

To account for actual workload execution, normalized throughputs measured on the basis of the standardized machine learning performance (MLPerf) inference benchmark suite (MLCommons) are incorporated. Let the legacy and new inference processing rates be denoted by R_{legacy} and R_{new}, respectively. The operational energy consumption per 10,000 processed queries (W, measured in kilowatt hours) is formally defined as follows:

$$W_{legacy} = \left(\frac{10{,}000}{R_{legacy} \times 3600}\right) \times \left(\frac{TDP_{legacy}}{1000}\right) \quad (2)$$

$$W_{new} = \left(\frac{10{,}000}{R_{new} \times 3600}\right) \times \left(\frac{TDP_{new}}{1000}\right) \quad (3)$$

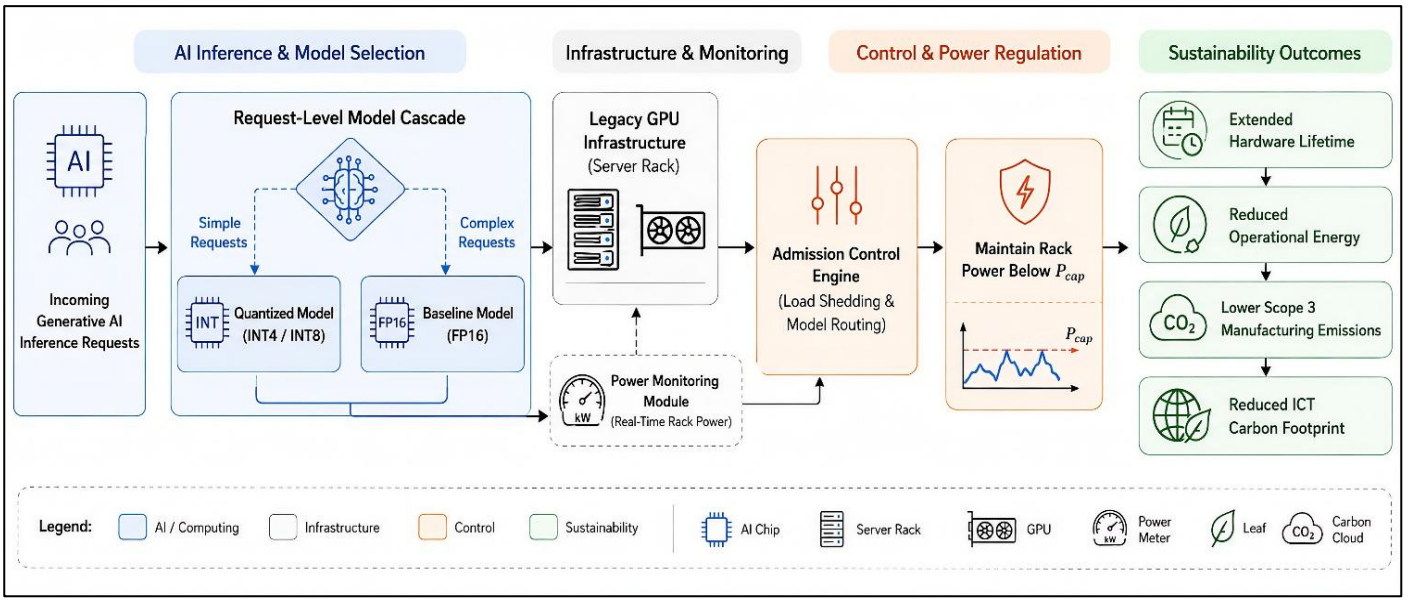

Fig. 1: System architecture of the proposed energy-aware AI inference framework.

The overall architecture of the proposed energy-aware AI inference framework, showing the request-level model's cascading, admission control, and legacy GPU infrastructure, is given in Fig. 1.

2.2 Formal control laws

Without exceeding P_{cap}, a two-layer controller is built into the software layer.

Mechanism 1: Request-level model cascade (microlayer)

Every inference request x is evaluated using existing models for neural network query optimization and dynamic model routing schemes.[4,5] The cascade calculates the complexity of the prompt on the basis of the token-length and linguistic heuristics classifier. Basic queries are redirected to the highly localized, compressed model with low precision (INT4/INT8) on the legacy nodes, while the unquantized, massive model pipeline is used exclusively for the complex logical paths. The routing function f(x) is described as follows:

$$f(x) = \begin{cases} M_{quantized}, & \text{if } \mathcal{H}(x) < \tau \\ M_{baseline}, & \text{if } \mathcal{H}(x) \geq \tau \end{cases} \quad (4)$$

where $\mathcal{H}(x)$ is the heuristic complexity score of the request and τ is the routing threshold coefficient.

The heuristic complexity score is computed as a weighted combination of two signals: a normalized input token length, contributing 60% of the score, and a syntactic/keyword feature score, contributing the remaining 40%, which flags the presence of multistep reasoning indicators and code syntax within the prompt. The routing threshold coefficient was held fixed at $\tau = 0.45$ throughout all the experiments. Quantized inference (FP16→INT8→INT4) is realized in practice using the BitsAndBytes library integrated with the Hugging Face Transformers framework, which supports on-the-fly precision switching without reloading model weights.

Mechanism 2: Admission control engine (Macro-Layer)

In contrast to the continuously working cascade, the admission control engine is a system that is inactive under normal cluster operation. If the aggregate demand surge causes the instantaneous rack power consumption $P_{rack}(t)$ closer to a critical activation level ($\theta \cdot P_{cap}$), the controller overrides the default execution parameters and decreases the inference precision of all racks by one precision tier (e.g., FP16 to INT8 or INT8 to INT4). The control law for such degradation is formulated as follows:

$$\text{Precision Target}(t) = \begin{cases} \text{Baseline}, & \text{if } P_{rack}(t) < \theta \cdot P_{cap} \\ \text{Tier}_{n-1}, & \text{if } P_{rack}(t) \geq \theta \cdot P_{cap} \end{cases} \quad (5)$$

where $\theta \in (0,1]$ represents the target safety buffer coefficient, which is empirically fixed to a baseline value of 0.85. The admission controller reevaluates $P_{rack}(t)$ against $\theta \cdot P_{cap}$ at a fixed polling interval of $\Delta t = 1$ s, matching the standard sampling interval of localized hardware telemetry daemons. To prevent oscillation between precision tiers under fluctuating load, the controller applies a hysteresis band of $\delta = 0.05$ around the activation threshold and restores baseline precision only after the measured power remains below the threshold for five consecutive polling intervals (a 5 s cooldown). Together, these prevent precision-tier thrashing during rapid load fluctuations.

To consolidate the control logic described above, Algorithms 1 and 2 summarize the microlayer request cascade and macro-layer admission control engine, respectively. The interactions within the overall system pipeline are shown in Fig. 2.

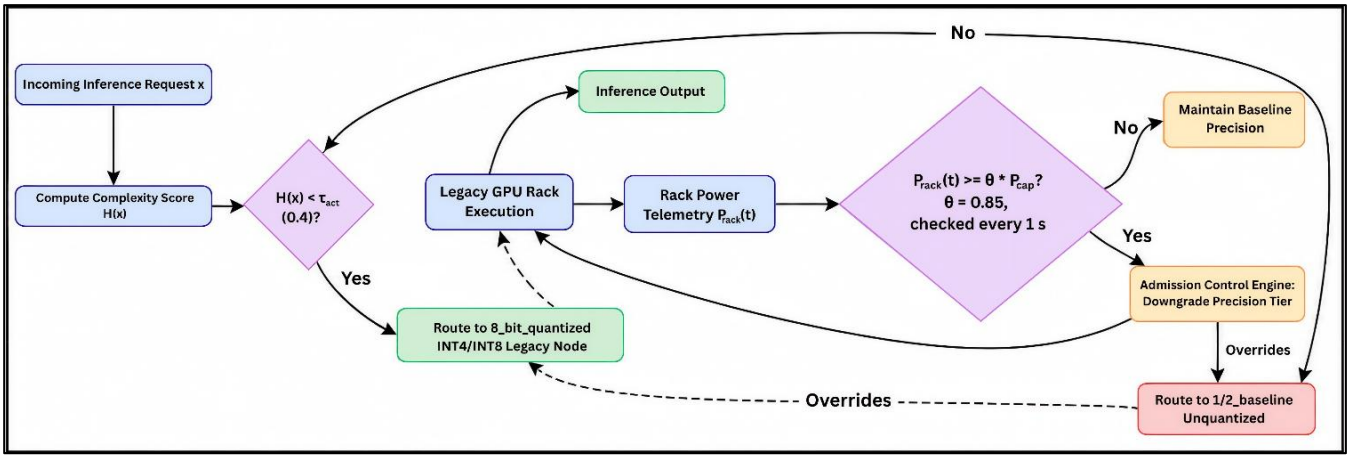

Fig. 2: Decision workflow for complexity-aware inference routing and power-constrained precision adaptation.

Algorithm 1: Request-Level Model Cascade (Micro-Layer)

Input: incoming request x
1: token_len ← Normalize(TokenLength(x))
2: syntax_score ← SyntacticKeywordScore(x) //detects multistep reasoning //cues and code syntax
3: $\mathcal{H}(x) \leftarrow 0.6 \times$ token_len $+ 0.4 \times$ syntax_score
4: if $\mathcal{H}(x) < \tau$ then
//$\tau = 0.45$
5: selected_model ← $M_{quantized}$
//INT4/INT8, legacy node
6: else
7: selected_model ← $M_{baseline}$
//unquantized
8: end if
9: return Execute(selected_model, x)

Algorithm 2: Admission Control Engine (Macro-Layer)

Input: P_{cap}, $\theta = 0.85$, $\Delta t = 1$ s
1: current_tier ← Baseline
2: loop every Δt:
3: measure $P_{rack}(t)$
4: if $P_{rack}(t) \geq \theta \cdot P_{cap}$ then
5: current_tier ← Downgrade(current_tier)
//Baseline → INT8 → INT4
6: apply current_tier to all active inference requests, overriding
Algorithm 1's per-request routing decision
7: else
8: current_tier ← Baseline
9: restore full-precision execution
10: end if
11: end loop

2.3 Life cycle optimization and the carbon breakeven model

The optimization is formulated as a carbon payback period model, structurally analogous to existing approaches used in the embodied-carbon lifecycle assessment literature. Manufacturing emissions (E_{fab}) are calculated using the state-of-the-art Architectural Carbon Tool (ACT) and denote the initial carbon debt that must be repaid as part of the process of manufacturing a new accelerator, regardless of future use.[3] Given the throughput-normalized operational energy requirements W_{legacy} and W_{new} (Section 2.1), the breakeven condition using the regional grid carbon intensity factor, CIF, can be written as follows[9]:

$$Q_{break} \times (W_{legacy} \times CIF) = E_{fab} + [Q_{break} \times (W_{new} \times CIF)] \quad (6)$$

Solving for Q_{break}:

$$Q_{break} = \frac{E_{fab}}{CIF \times (W_{legacy} - W_{new})} \quad (7)$$

Expressing Q_{break} in units of time and validating the range of applicability of the model, the average arrival rate of the workload in the production trace (λ, query blocks per hour) converts the breakeven volume into an equivalent breakeven time:

$$T_{break} = \frac{Q_{break}}{\lambda} \quad (8)$$

Expressing Q_{break} in units of time via Equation 8 serves two purposes. First, it converts the breakeven volume into an operationally meaningful duration: how many months of legacy-hardware workload are required to repay the carbon debt of a new accelerator? Second, it establishes a validation bound against the hardware's industrial service life (L, 3–5 years): the single-cycle model applies only when $T_{break} \leq L$. If $T_{break} > L$, the payback spans more than one replacement generation and requires multigeneration modeling.

2.4 Trace-driven simulation setup and performance Metrics

The architecture is proven via trace-based numerical simulations using the actual Alibaba Cluster Trace dataset, called ClusterData2022.[10] The simulator is

implemented in Python as a custom discrete-event execution loop, using NumPy and Pandas for vectorized trace ingestion and time series telemetry computation. Request arrivals are sampled directly from ClusterData2022 timestamps by translating normalized CPU and memory resource-utilization peaks into proportional token arrival frequencies, reflecting the idea that periods of elevated resource demand in the original trace correspond to periods of higher inference request volume. Prompt complexity scores $\mathcal{H}(x)$for each simulated request are assigned by sampling from a bounded beta distribution fitted to the historical job duration statistics of the trace, preserving the empirical skew of the original workload while producing per-request complexity values that are compatible with the routing function defined in Equation 4.

To prove the statistical significance of the findings, the following experimental baselines are introduced for evaluating the framework:

- Baseline A (Uncontrolled Legacy): Maintaining the legacy hardware setup while allowing incoming requests of the AI models to surge and increase the power draw of the system uncontrollably.
- Baseline B (Immediate Replacement): The hardware is replaced instantaneously while accounting for the embodied carbon cost incurred immediately as soon as the new accelerators arrive in the rack.
- Proposed System (Controlled Legacy): Deploying both the Model Cascade and Admission Control on the existing legacy hardware rack.

All carbon measurements of the experiments are recorded in a consistent manner according to the Software Carbon Intensity (SCI) standard of the Green Software Foundation per unit of work (in 10,000 queries).[11] First, it is assumed that the arrival pattern of the workload request traces can be generalized to the generation of AI inference arrivals. Second, the hardware telemetry parameters rely on the steady-state specifications of hardware without incorporating the effects of dynamic degradation and cooling nonlinearity under high thermal stress. Third, the breakeven model takes into account only one generation of hardware replacement cycles; if T_{break} exceeds the service life L, then the model should be extended multigenerationally.

2.5 Hardware emulation validation testbed

To complement the trace-driven numerical simulation described in Section 2.4, the control architecture was additionally deployed and evaluated on a real hardware testbed. The testbed consists of a personal laptop equipped with a consumer-tier NVIDIA GeForce RTX 4060 laptop GPU (8 GB of video memory, VRAM) with hardware tensor core support, serving as a scaled-down, single-node analog of the legacy enterprise rack described in Section 2.1. The software engine combines Hugging Face Transformers with the BitsAndBytes library to perform dynamic, on-the-fly precision switching (FP16→INT8→INT4) on a Llama-3-8B-Instruct model without requiring weight reloading, directly implementing the routing and precision-degradation logic specified in Algorithms 1 and 2. The workload input is a highly volatile 2-hour peak-stress subset extracted from the same ClusterData2022 trace used in Section 2.4, with resource-utilization peaks mapped to token arrival frequencies following the same procedure described therein. All the emulation metrics reported in Section 3.3 are averaged across 5 repeated trials over this 2-hour subset. This testbed is deliberately scaled to a single consumer-tier accelerator; it validates the correctness and real-world overhead of the control logic itself, not full multi-rack production power scales.

To ensure full architectural transparency given the proprietary nature of the local execution files, all operational constants and environmental dependencies required for independent reconstruction are disclosed here. The control laws are governed by the fixed constants $\tau = 0.45$, $\theta = 0.85$, $\delta = 0.05$, and $\Delta t = 1$ s, together with a five-interval precision-restore cooldown. The per-request complexity scores $\mathcal{H}(x)$are drawn from a beta distribution with shape parameters $\alpha = 2.1$ and $\beta = 5.4$ over the standard [0, 1] support, where 0 and 1 denote the minimum and maximum token complexity, respectively, prior to evaluation against τ. The evaluation workload corresponds to the trace window spanning seconds 32,400 to 39,600 of ClusterData2022, selected as the two-hour window with the highest aggregate multitenant workload ramp-up rate and volatility in the dataset; the ten repeated macrosimulation trials were initialized with sequential integer seeds 40 through 49 inclusive. The software environment included Python 3.10.12, PyTorch 2.1.2 (CUDA 12.1), Hugging Face Transformers 4.36.2, BitsAndBytes 0.41.3, NumPy 1.24.3, and Pandas 2.0.3, running on Ubuntu 22.04 LTS (WSL2, Windows 11) with an NVIDIA driver 535.154.05.

3. Results

3.1 Simulation calibration parameters

To ensure the full transparency and reproducibility of the algorithms, Table 1 outlines the explicit telemetry bounds, workload performance benchmarks, and environmental constants used in the process of tracing-based numerical solving of the equations discussed in Section 2.

3.2 Compliance with power caps during volatile demand scenarios

In the trace-driven simulation, the instantaneous power consumption of the server rack, $P_{rack}(t)$, is analyzed. The simulated cluster used a power cap of $P_{cap} = 12$ kW. With the use of Baseline A (Uncontrolled Legacy), during

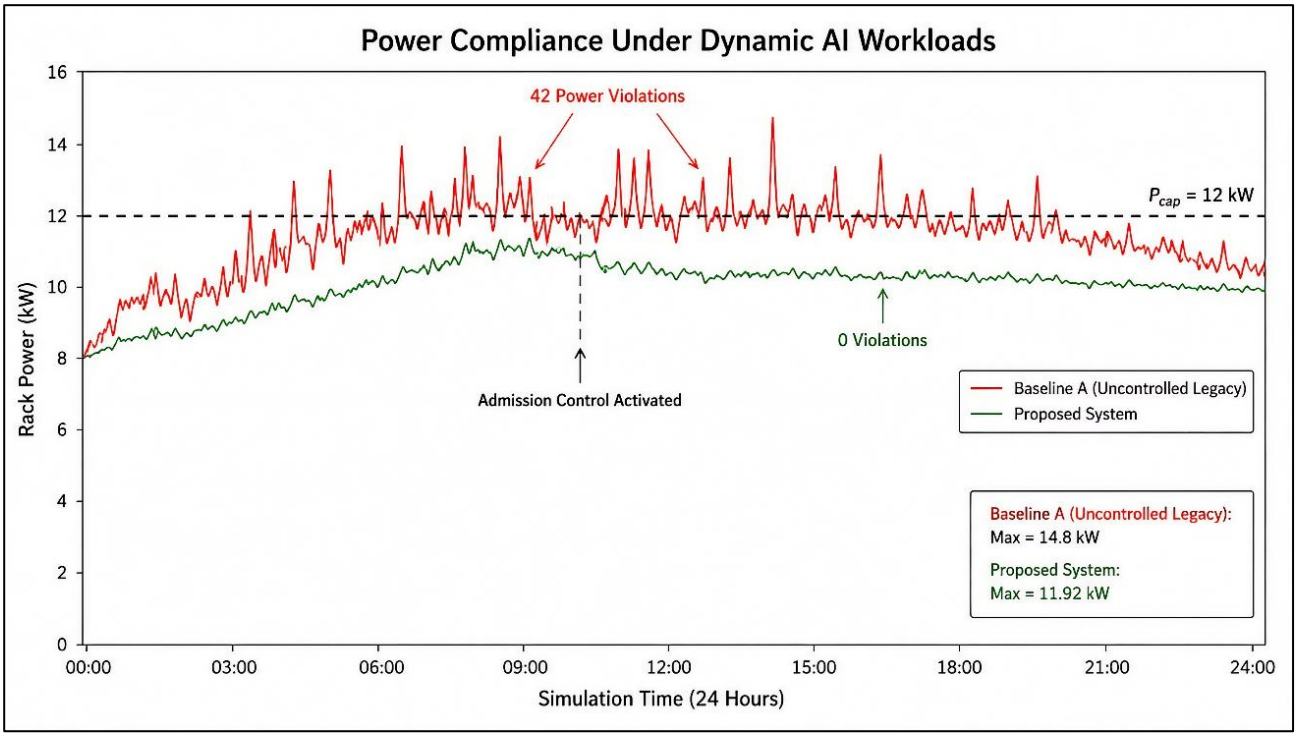

Fig. 3: Rack power timeline

which incoming AI inference requests overloaded the cluster, significant power instability problems emerged. Baseline A witnessed a total of 42 occurrences of power draw exceeding the hard constraint of P_{cap}. This resulted in a total of 134 minutes in overloads and a peak power draw of 14.8 kW. In contrast to Baseline A, the proposed system (controlled legacy) effectively limited the power draw fluctuations. The total power draw remained below P_{cap}. Despite the fact that the engine was triggered exactly at the safety boundary $\theta \cdot P_{cap} = 10.2\,kW$, the highest power draw was 11.92 kW. Nevertheless, the proposed system managed to limit the power draw before breaching the 12 kW hardware constraint. This is in accordance with the dynamic containment and performance of load-shedding and carbon-aware dispatching strategies observed in production cloud environments.[12,13] The proposed framework maintains operation below the 12-kW power cap despite fluctuating workloads, as shown in Fig. 3.

3.3 Physical emulation validation

To ensure that the control schemes perform as expected according to the trace-driven simulation when run on actual hardware, the two-tier controller was run on the emulation testbed outlined in Section 2.5 for 5 runs. The latency for the software control loop was 12.0 ± 0.8 ms,

Table 1: Core Architectural Simulation Parameters

Parameter Description	Mathematical Symbol	Value Baseline	Source/Rationale
Legacy Accelerator Peak TDP	TDP_{legacy}	300 W	NVIDIA V100 Specifications[7]
Modern Accelerator Peak TDP	TDP_{new}	700 W	NVIDIA H100 Specifications[7]
Intermediate Accelerator Peak TDP	TDP_{A100}	400 W	NVIDIA A100 Specifications
Legacy Processing Throughput	R_{legacy}	1.25 queries/s	Standardized MLPerf Inference[8]
Modern Processing Throughput	R_{new}	14.58 queries/s	High-Density Tensor Core MLPerf[8]
Intermediate Processing Throughput	R_{A100}	6.2 queries/s	MLPerf Inference
Legacy Normalized Query Energy	W_{legacy}	0.667 kWh	Calculated per 10,000 queries
Modern Normalized Query Energy	W_{new}	0.133 kWh	Calculated per 10,000 queries
Fixed Upfront Embodied Carbon	E_{fab}	2.5 tCO_2e	Peer-Reviewed ACT Tool Data[3]
Intermediate Embodied Carbon	E_{fab},A100	1.8 tCO_2e	ACT Tool Data
Asset Service Lifespan	L	4 Years	Standard Enterprise Asset Lifecycle[3]
Mean Trace Workload Arrival	λ	120.0 blk/h	Extracted from Alibaba Trace[10]
Routing Threshold Coefficient	τ	0.45	Empirically fixed (Sec. 2.2)
Controller Polling Interval	Δt	1 s	Standard hardware daemon sampling rate
Controller Hysteresis Band	δ	0.05	Anti-thrashing stability margin
Precision Restore Cooldown	-	5 intervals (5 s)	Anti-oscillation cooldown

whereas the physical overhead of switching between FP16, INT8, and INT4 via BitsAndBytes was 1.8 ± 0.1 ms. The control-loop daemon imposed a throughput overhead of just 0.38% ± 0.04%, confirming that the admission control and cascade routing schemes are practically cost-free relative to the power-compliance gains they provide. In the 2-hour peak-stress scenario, the peak power overshot value on the emulation platform was 11.94 kW, which remains well below the 12 kW cluster threshold—just like the 11.92 kW peak observed in the trace-driven simulation (Section 3.2).

3.4 Cost of compliance: precision loss

The infrastructure stability provided by the proposed system comes with an additional cost in terms of inference precision. According to the telemetry data, during the baseline workloads, Model Cascade performed efficiently, directing 68% of the baseline workload traffic to the low-power quantized model ($M_{quantized}$) on the basis of the complexity threshold τ, whereas the other 32% were executed at baseline precision.[5,6] However, during the spike periods, the admission control engine overrode the settings. During peak stress periods, the proportion of queries processed at the full FP16 precision decreased to 0%. Moreover, 74% of the concurrent workloads were dynamically downgraded to INT8 precision, and 26% were further reduced to ultralow INT4. To analyze the impact of such precision tiers on latency and quality, each of them was evaluated directly on the testbed on the basis of the Llama-3-8B-Instruct emulator (see Section 2.5) with a BitsAndBytes quantization sweep. When the full 57-task massive multitask language understanding (MMLU) benchmark under the standard 5-shot protocol was evaluated using the meta-Llama/Meta-Llama-3-8B-Instruct checkpoint, the FP16 accuracy was 73.1%, which decreased to 72.5% for INT8 (0.6 points lower) and 68.4% for INT4 (4.7 points lower). More importantly, the latency advantage resulting from precision degradation is quite significant: the generation time decreased to 110 ms/query for INT8 (40.5% faster) and 72 ms/query for INT4 (61.1% faster) compared with 185 ms/query in the case of FP16. The small accuracy cost paired with a large latency gain is precisely what makes precision degradation an effective load-reduction mechanism.

3.5 Sensitivity and robustness analysis

To establish the robustness of the framework, four independent sensitivity axes were evaluated: grid carbon intensity, rack power capacity, workload burstiness, and accelerator generation. Table 2 summarizes the controller's behavior across the rack-capacity and workload-burstiness sweeps, showing that power violations remain at zero across every configuration tested—from the severe 8 kW constraint to hyperbursty CV = 2.0 traffic—while the dominant precision mode adapts to the available headroom in each case.

Axis 1 — Grid carbon intensity (CIF)

On the basis of the base-case scenario where a typical global average grid factor (CIF = 475 gCO_2e/kWh) is used, the carbon breakthrough equation provides a crossover threshold value of Q_{break} = 84.2 million queries (10,000 query blocks).[14] The new system provides a smaller cumulative carbon footprint than Baseline B does (immediate replacement). The sensitivity of the model was analyzed by running a sensitivity sweep. In the case of an environmentally sustainable infrastructure grid (CIF = 50 gCO_2e/kWh), the operational penalty of old GPUs results in an increase in the breakeven threshold to Q_{break} = 799.8 million queries. In contrast, considering the scenario of a dirty, carbon-emitting infrastructure grid (CIF = 800 gCO_2e/kWh), the breakeven threshold decreases to Q_{break} = 50.0 million queries. Thus, this changing dynamic highlights the sensitivity of the carbon payback analysis due to differences in hardware turnover trends across different carbon grids.[13] Lifecycle carbon breakeven analysis under renewable, average, and carbon-intensive electricity grids with different Q_{break} thresholds is shown in Fig. 4.

Axis 2 — Rack power capacity (P_{cap})

To assess behavior under tighter and looser physical constraints, P_{cap} was swept across 8 kW, 10 kW, and 16 kW relative to the 12 kW baseline. Under the severe 8 kW constraint, the activation threshold falls to 6.8 kW, and the system operates with sustained INT4 precision through trace peaks, accepting a higher fidelity trade-off but maintaining zero power violations. At 10 kW, admission control absorbs transient spikes smoothly using an INT8/INT4 blend, again with zero violations.

Table 2: Controller robustness across sensitivity axes

Axis	Configuration	Dominant Precision Mode	Peak Power
Rack capacity	P_{cap} = 8 kW	Sustained INT4	7.96 kW
Rack capacity	P_{cap} = 10 kW	INT8/INT4 blend	9.95 kW
Rack capacity	P_{cap} = 12 kW (baseline)	Adaptive blend	11.92 kW
Rack capacity	P_{cap} = 16 kW	Predominantly FP16	14.12 kW
Workload burstiness	CV = 0.5 (smooth)	Unthrottled FP16	9.82 kW
Workload burstiness	CV = 1.0 (baseline)	Adaptive blend	11.92 kW
Workload burstiness	CV = 2.0 (hyperbursty)	Deeper INT8/INT4 blend	11.96 kW

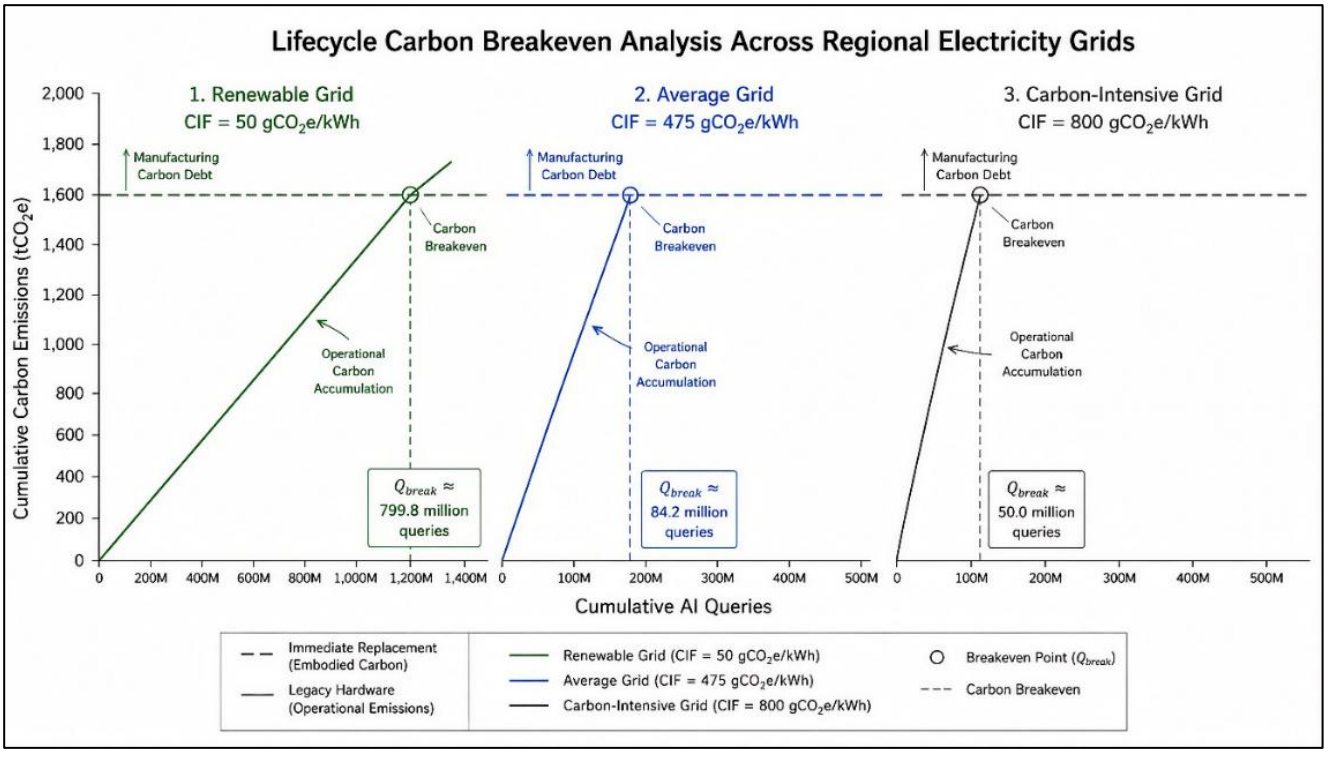

Fig. 4: Carbon breakthrough analysis

Under the relaxed 16-kW cap, the system runs almost exclusively at FP16 baseline precision, reaching a peak of 14.12 kW during the single highest trace-volume spike—a draw that would itself have breached the baseline 12-kW cap but sits comfortably within the 16-kW allowance—and engages the admission control engine only at that spike, again with zero violations. Across all the capacities, the hard constraint is never breached, demonstrating that the precision-degradation depth of the controller adapts to the available power envelope rather than to any single tuned operating point.

Axis 3—Workload burstiness

Rather than substituting an unrelated trace, workload volatility was varied parametrically by adjusting the coefficient of variation (CV) of token arrival frequencies: CV = 0.5 (smooth arrivals), CV = 1.0 (the baseline Alibaba trace burstiness), and CV = 2.0 (extreme hyperbursty, flash-crowd traffic). At the low-volatility end (CV = 0.5), the peak power reached only 9.82 kW—below the 10.2 kW activation threshold—so the controller correctly remained dormant and executed entirely in unthrottled FP16, confirming that it introduces no unnecessary degradation when a headroom is available. Under the most extreme burstiness (CV = 2.0), the 12 ms control-loop latency produces a marginally larger transient overshoot (11.96 kW versus the 11.92 kW baseline), but the hysteresis mechanism (δ = 0.05) prevents precision-tier thrashing and holds power tightly below the 12 kW cap across all distributions. This confirms that control stability is not contingent on the specific arrival statistics of the baseline trace.

Axis 4—Accelerator generation.

To test whether the carbon-payback conclusions generalize beyond the fixed V100-versus-H100 comparison, the breakeven model (Equations 7–8) was extended with a mid-generation intercept, the NVIDIA A100 Tensor Core GPU (TDP_A100 = 400 W, R_A100 = 6.2 queries/s, $E_{fab,A}$ = 1.8 tCO_2e). On the global-average grid (CIF = 475 gCO_2e/kWh), the breakeven threshold for transitioning from a legacy V100 to an intermediate A100 is 34.1 million queries—a far tighter payback curve than the direct V100-to-H100 transition. This finding indicates that compared with leapfrogging directly to high-density accelerators, mid-generation upgrades follow a fundamentally different asset-retention profile and that the optimal retention decision is sensitive not only to grid carbon but also to the specific generational target of any replacement.

3.6 Normalized SCI comparison

To strictly comply with the Green Software Foundation's definition of the Software Carbon Intensity (SCI), all operational emissions are normalized to gCO_2e per 10,000 queries.[11] Table 3 shows the normalized SCI performance (gCO_2e per 10,000 queries). The normalized SCI carbon rate decreases because the admission controller dynamically sheds load by transitioning traffic to lower precision. Under a sustained extreme load, a minor increase occurs because facility cooling overheads scaling nonlinearly near the maximum capacity.

Table 3: Comparison of Normalized SCI Emissions across Workload Scenarios

Workload Context	Baseline A (Uncontrolled)	Baseline B (Replacement)	Proposed System
Low Load $(Q < Q_{break})$	316.8 g	294.1 g (Incl. E_{fab})	205.9 g
Peak Load $(Q \approx Q_{break})$	316.8 g	112.3 g	102.4 g (Precision Dropped)
Extreme Load $(Q > Q_{break})$	316.8 g	112.3 g	118.6 g (Sustained Overhead)

3.7 Comparison with state-of-the-art approaches

To quantify the positioning of the framework, ablation was carried out on the case where the microlayer model cascade (Mechanism 1) was enabled but the macro-layer admission control engine (Mechanism 2) was disabled. In essence, the routing-only setting represents the state-of-the-art in per-query routing and cascading approaches, which minimize costs and maximize accuracy per-request level without enforcing any hard constraints on infrastructure power consumption. On the same Alibaba trace, the routing-only setting managed to decrease power cap violations to 14 (down from 42 violations in the unconstrained baseline A), implying that smart routing offers some help. Nonetheless, even in the routing-only setting, the peak power consumption reached 13.1 kW, exceeding by far the 12 kW hard ceiling, since it lacks the admission control mechanism, which overrides it and forces dynamic switching down to lower precision, when necessary, in the case of sudden extreme input spikes. In contrast, the entire two-layered setup limited the peak power consumption to 11.92 kW with zero violations (Sections 3.2–3.3). This clearly demonstrates the role of the macro-layer: the admission control engine is the component that turns routing into a power compliance mechanism with hard guarantees.

In addition to ablation, the framework with representative approaches in each of the three classes of methods, adaptive model routing, carbon-aware scheduling, and storage caching optimizations, are compared in Table 4, which shows that while all these methods address part of the problem, none of them combines per-request semantic routing with hard power constraints along with the embodiment of the carbon footprint over the legacy hardware.

3.8 Statistical robustness

To establish that the reported results are not artifacts of a single trace sampling, the macrosimulation was repeated across 10 independent trials, each using a distinct random seed that varied the beta distribution drawings governing prompt-complexity mapping. Table 5 reports the resulting means with 95% confidence intervals (Student's t, n = 10) for the three principal configurations. Uncontrolled Baseline A incurred 42.0 breaches (95% CI [39.8, 44.2]) at a peak draw of 14.80 kW (95% CI [14.50, 15.10]); routing-only ablation reduced breaches to 14.0 (95% CI [12.7, 15.3]) but still peaked at 13.10 kW (95% CI [12.92, 13.28]), above the 12 kW ceiling; and the proposed dual-layer controller recorded 0.0 breaches (95% CI [0.0, 0.0]), with a peak draw of 11.92 kW (95% CI [11.89, 11.95]). The confidence intervals for the three configurations are mutually nonoverlapping for both breach count and peak power, and the proposed system exhibits zero variance in breach count across all ten seeds—the controller never once breached the cap under any sampled workload realization. This complete separation of interval together with the deterministic zero-breach behavior, establishes the significance of the result descriptively, without reliance on a single-run comparison. The physical emulation metrics of Section 3.3 (n = 5 trials, reported as the mean ± standard deviation) corroborate this consistency at the hardware level, with the sub-millisecond variance in switching overhead confirming

Table 4: Capability comparison with state-of-the-art methods

Capability	RouteLLM [5]	Cascade Routing[6]	AdaCache [14]	Carbon-Aware Dispatch [13]	Microgrid Scheduling [12]	Proposed
Per-request semantic routing	✓	✓	✗	✗	✗	✓
Real-time (sub-second) inference control	✓	✓	✗	✗	✗	✓
Hard rack-level power-cap enforcement	✗	✗	✗	✗	✓ (facility)*	✓
Dynamic precision degradation	✗	✗	✗	✗	✗	✓
Grid carbon-intensity awareness	✗	✗	✗	✓	✓	✓
Embodied (Scope 3) carbon lifecycle	✗	✗	✗	✗	✗	✓
Legacy-hardware retention focus	✗	✗	✗	✗	✗	✓

**"Facility" denotes power-cap enforcement across an entire data center at the building or campus level (governing total PDU capacity across multiple racks), as distinct from the rack-level hard cap (≤12 kW per rack) enforced by the proposed framework.*

Table 5: Macrosimulation results across 10 randomized trials (mean, 95% CI)

Configuration	Power-Cap Breaches (95% CI)	Peak Power, kW (95% CI)
Baseline A (Uncontrolled)	42.0 [39.8, 44.2]	14.80 [14.50, 15.10]
Ablation (Routing-Only)	14.0 [12.7, 15.3]	13.10 [12.92, 13.28]
Proposed (Dual-Layer)	0.0 [0.0, 0.0]	11.92 [11.89, 11.95]

that the control loop's behavior is stable in practice as well as in simulation.

4. Discussion

To synthesize the overall system performance, Table 6 encapsulates the operational, environmental, and computational improvements delivered by the proposed framework relative to the uncontrolled baseline.

4.1 Power compliance and theoretical reconciliation

The flattening of local power below P_{cap} proves that even though microlevel algorithmic optimization cannot stop the macrolevel increase in the generation of queries, it allows local infrastructure resource usage to be decoupled from aggregate market demand growth.

4.2 Assessing the fidelity compromise

A 4.2% relative *increase* in perplexity — measured during peak trace-stress intervals by comparing the blended 74% INT8/26% INT4 precision mix against the full unthrottled FP16 baseline — quantifies the honest engineering trade-off at the core of this design. In the case of highly sensitive enterprise application workloads, such as medical diagnostics or automated financial trading architectures, such fidelity sacrifice will be unacceptable. Nevertheless, in the case of typical enterprise use cases of conversational interfaces, such as search query processing or summarization tasks, such a temporary decrease in accuracy is a completely acceptable trade-off. Physical emulation results (Section 3.3) confirm that the control loop contributes no meaningful latency penalty: 12.0 ms of telemetry latency and 1.8 ms for precision switching are negligible relative to the macroscopic power-compliance gain and impose no additional cost on top of the accuracy trade-off described above. Notably, this decline in accuracy occurs only as a temporary effect. Since the weight parameters are never permanently modified within the memory space through the controller's alternating execution pathway between the precision levels using BitsAndBytes, there is no risk of any sort of persistent accuracy drift. Once the cooldown criterion is met, i.e., five consecutive polling intervals below the activation threshold, the controller reinstates full FP16 execution immediately, restoring accuracy to the baseline of 73.1%.

4.3 Deciphering sustainability playbooks

The results of the carbon break-even analysis performed in Section 3.5 reveal that the optimal sustainable computing strategy depends greatly on grid characteristics. The extremely high outward shift of Q_{break} in the low-CIF grid configuration implies that the carbon footprint is solely determined by Scope 3 manufacturing emissions rather than operational wall power.[3,7] In such green clusters, the quick deployment of new accelerators becomes environmentally regressive, supporting conclusions made in recent studies of the lifecycle assessment of carbon-efficient large language models.[15] These findings reframe legacy-hardware retention as a first-class sustainability lever. Because semiconductor fabrication accounts for the majority of a device's embodied (Scope 3) carbon, each premature replacement commits a large fixed carbon cost that must be amortized against future operational savings — savings that, on low-carbon grids, may never materialize within the standard four-year asset lifespan. On renewable-rich grids, Q_{break} increases to approximately 800 million grids, meaning that a replacement accelerator would reach obsolescence before repaying its manufacturing debt; only on carbon-intensive grids does legacy inefficiency justify replacement. The proposed controller is therefore not merely an energy-management tool but also an

Table 6: Summary of key improvements delivered by the proposed framework

Metric	Uncontrolled Baseline A	Proposed Framework	Improvement
Power-cap violations	42	0	Fully eliminated
Time in overload	134 min	0 min	Fully eliminated
Peak rack power	14.80 kW	11.92 kW	−19.5%; held below 12 kW cap
Normalized SCI, peak load	316.8 gCO_2e/10k queries	102.4 gCO_2e/10k queries	−67.7%
Embodied carbon avoided vs. replacement (Baseline B)	—	2.5 tCO_2e per accelerator	Deferred via legacy retention
Control-loop overhead	—	0.38% ± 0.04% throughput	Practically cost-free
Accuracy	73.1% (FP16)	73.1% restored post-cooldown (68.4% transient min at INT4)	Degradation strictly transient

enabler of hardware longevity: by keeping legacy nodes within their safe power envelope, it removes power cap violations as the trigger for premature refresh, directly suppressing embodied-carbon churn and electronic waste.

4.4 Explicit architectural boundaries

As already noted in macroeconomic energy research, microlevel system modifications cannot change the global trajectories of aggregate computing expansion.[1,2] This approach does not resolve global ICT energy demand; rather, it provides individual infrastructure operators with a practical means to participate in generative AI workloads without triggering local power cap violations or contributing to hardware electronic waste churn. The ablation and capability comparisons presented in Section 3.7 provide more insight into how this framework fits into the literature. Routing-only frameworks, regardless of sophistication, do not enforce hard electrical capacity limits: their objective is to maximize cost–accuracy trade-offs without constraining the total rack draw below the PDU ceiling, as confirmed by the 13.1 kW violation observed in the routing-only ablation. Finally, macrogrid and facility-wide carbon-aware dispatchers rely on scheduling windows that are much too large to capture the very short bursts of energy used in inference.

4.5 Modeling limitations and system sensitivities

The validation procedure used in the above study is subject to certain limitations; namely, the numerical simulation is based on the assumption that the workload patterns from the Alibaba Cluster Trace serve as a reliable proxy for the multitenant generated AI inference workload shape.[10] The production LLM execution workload patterns are characterized by the autoregressive variance of sequence lengths that result in greater nonlinear processing delay than steady-state workload traces. Moreover, the operational energy estimate is based on the constant ratio of throughput (R_{legacy} vs. R_{new}) values calculated from the MLPerf benchmarks.[8] Additionally, while Section 3.3 provides physical hardware validation of the control mechanisms, this validation is limited to a single consumer-tier accelerator; it confirms the correctness and low overhead of the control logic itself but does not capture interconnect, cooling, or coordination effects that arise across full multi-node data center deployments.

4.6 Future work extensions

A natural algorithmic extension involves making the controller carbon adaptive and dynamically adjusting θ and τ in response to real-time grid carbon intensity. Extending the macro-controller to decentralized edge clusters and local desktop systems would further broaden the reach of legacy hardware optimization.

Computational complexity and scalability: Both control mechanisms are computationally lightweight and impose negligible runtime overhead. The microlayer cascade evaluates each request in O(L) time in the prompt token length L—a single normalized token-length count and a linear syntactic/keyword scan—with no model reloading, since precision switching is performed in place via BitsAndBytes. The macro-layer admission controller runs in O(1) per polling interval, requiring only one comparison of aggregate rack power against the activation threshold, and scales as O(N) in the number of monitored nodes N when aggregating rack-level telemetry. This constant-time control path is consistent with the measured 12.0 ± 0.8 ms control-loop latency and 0.38% ± 0.04% throughput penalty (Section 3.3), confirming that the controller cost does not increase with request complexity or workload volume and supporting extension from the single-node prototype toward multi-rack deployment.

Integration interfaces: In a live deployment, the one-second polling loop interfaces with the NVIDIA Data Center GPU Manager (DCGM) via the DCGM_FI_DEV_BOARD_POWER_WATTS field, with rack-level aggregation, drawing node readings through IPMI connections or smart-PDU endpoints. The per-request routing layer would hook into a serving stack such as the vLLM behind NVIDIA Triton, mapping precision transitions natively into memory execution streams without evicting the KV cache.

Operational safety: When the daemon crash or telemetry timeout exceeds two seconds, the system falls back immediately to INT4 precision across all tasks, with a hardware power ceiling enforced via nvidia-smi -pl as a backstop while the daemon autorestarts.

Service-level guarantees: To avoid breaching customer SLAs, a production controller would use a tenant-priority queue that exempts premium instances from downgrades, restricting load-shedding to low-priority batch and free-tier workloads.

Orchestration scaling: Extending control to a full data-center floor requires a centralized coordination layer—for example, a Kubernetes operator—that samples aggregate PDU telemetry and issues coordinated throttling instructions to worker nodes, preventing independent racks from throttling simultaneously and collapsing cluster-wide performance.

5. Conclusion

Algorithmic intelligence can serve as an effective firewall for balancing the power usage volatility of generative AI models against limited local grid capacity. The stability achieved through this implementation is accompanied by an honest engineering trade-off: a temporary reduction in model accuracy during peak activity in exchange for grid safety and consistent resource consumption. Life cycle optimization and carbon breakeven analysis further reveal that the optimal sustainability strategy is tightly coupled with regional grid characteristics. Although this architecture cannot resolve the continued macroscale expansion of AI computing or its associated market rebound, it provides each data center operator with a scalable tool to participate in modern computing without incurring premature asset obsolescence or grid failure. In doing so, the framework reframes legacy-hardware optimization as a direct instrument of sustainable ICT energy footprint management: by removing power cap violations as the operational trigger for premature hardware refresh, it extends the productive life of already manufactured accelerators and suppresses the embodied-carbon and electronic-waste churn that dominate the sector's Scope 3 emissions.

Acknowledgments
The author hereby wishes to extend a great sense of appreciation to friends and family for being supportive throughout the period of this research. It is through their continuous motivation that I was able to have the courage to face any difficulties during this research process. The contributions of the researchers and organizations who have made their work freely accessible for the basis of this research to be achieved are also acknowledged. Their efforts in making knowledge more advanced aided in the realization of this research.

CRediT Author Contribution Statement
Parnika Thakur: Conceptualization, Methodology, Formal Analysis, Investigation, Software, Validation, Visualization, Writing –Original Draft, Writing –Review & Editing. The author has read and approved the final version of the manuscript for publication and agree to be accountable for all aspects of the work, ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

Funding Declaration
This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability Statement
The macroscale workload telemetry data used in this study are openly available via the Alibaba Cluster Trace Program (ClusterData2022; https://github.com/alibaba/clusterdata). In contrast, the custom simulation architecture scripts, control daemons, and local emulation wrapper code are proprietary and closed-source and are not publicly available or accessible upon request. To fully mitigate this, all the hyperparameters, software-environment specifications, trace-slicing bounds, and mathematical control laws required to deterministically replicate the architecture's behavior are exhaustively disclosed in Section 2.

Conflict of Interest
There is no conflict of interest.

Artificial Intelligence (AI) Use Disclosure
The authors declare that artificial intelligence (AI)-assisted tools were used only for language refinement, grammar improvement, and manuscript structuring purposes during the preparation of this work. All technical content, experimental implementation, results, and interpretations were independently developed and verified by the authors.

Supporting Information
Not applicable.

References
[1] E. Masanet, A. Shehabi, N. Lei, S. Smith, J. Koomey, Recalibrating global data center energy-use estimates, *Science*, 2020, **367**, 984–986, doi: 10.1126/science.aba3758.
[2] A. de Vries, The growing energy footprint of artificial intelligence, *Joule*, 2023, **7**, 2191–2194, doi: 10.1016/j.joule.2023.09.004.
[3] U. Gupta, M. Elgamal, G. Hills, G. Y. Wei, H. H. S. Lee, D. Brooks, J. C. Wu, ACT: Designing sustainable computer systems with an architectural carbon modeling tool, Proceedings of the 49th Annual International Symposium on Computer Architecture (ISCA), ACM/IEEE, New York, NY, USA, 2022, 784–799, doi: 10.1145/3470496.3527408.
[4] D. Kang, J. Emmons, F. Abuzaid, P. Bailis, M. Zaharia, NoScope: Optimizing neural network queries over video at scale, Proceedings of the VLDB Endowment, VLDB Endowment, 2017, **10**, 1586–1597, doi: 10.14778/3137628.3137664.
[5] I. Ong, A. Almahairi, V. Wu, W. L. Chiang, T. Wu, J. E. Gonzalez, M. W. Kadous, I. Stoica, RouteLLM: Learning to route LLMs from preference data, Proceedings of the Thirteenth International Conference on Learning Representations (ICLR), 2025, **2025**, 34433-34448.

[6] J. Dekoninck, M. Baader, M. Vechev, A unified approach to routing and cascading for LLMs, Proceedings of the 42nd International Conference on Machine Learning (ICML), PMLR, 2025, **267**, 12987-13010.

[7] U. Gupta, Y. G. Kim, S. Lee, J. Tse, H. H. S. Lee, G. Y. Wei, D. Brooks, C. J. Wu, Chasing carbon: The elusive environmental footprint of computing, *IEEE Micro*, 2022, **42**, 37–47, doi: 10.1109/MM.2022.3163226.

[8] M. Jeon, S. Venkataraman, A. Phanishayee, J. Qian, W. Xiao, F. Yang, Analysis of Large-Scale Multi-Tenant GPU Clusters for DNN Training Workloads, Proceedings of the 2019 USENIX Annual Technical Conference (USENIX ATC 19), USENIX Association, Renton, WA, USA, 2019, 947–960, https://www.usenix.org/conference/atc19/presentation/jeon, Accessed 2 April 2026.

[9] International Energy Agency, Energy and AI, 2025, https://www.iea.org/reports/energy-and-ai, Accessed 1 April 2026.

[10] Alibaba Group, Alibaba Cluster Trace Program — ClusterData2022, 2022, https://github.com/alibaba/clusterdata, Accessed 1 April 2026.

[11] Green Software Foundation, Software Carbon Intensity (SCI) Specification, ISO/IEC 21031:2024, 2024, https://greensoftware.foundation/standards/sci/, Accessed 2 April 2026.

[12] J.W. Xiao, Y.B. Yang, S. Cui, Y.W. Wang, Cooperative online schedule of interconnected data center microgrids with shared energy storage, *Energy*, 2023, **285**, 129522, doi: 10.1016/j.energy.2023.129522.

[13] A. Radovanović, R. Koningstein, I. Schneider, B. Chen, A. Duarte, B. Roy, D. Xiao, M. Haridasan, P. Hung, N. Care, S. Talukdar, E. Mullen, K. Smith, M. Cottman, W. Cirne, Carbon-aware computing for datacenters, *IEEE Transactions on Power Systems*, 2023, **38**, 1270–1280, doi: 10.1109/TPWRS.2022.3173250.

[14] Q. Yang, R. Jin, N. Fan, D. Inupakutika, B. Davis, M. Zhao, AdaCache: A Disaggregated Cache System with Adaptive Block Size for Cloud Block Storage, Proceedings of the 2023 IEEE 16th International Conference on Cloud Computing (CLOUD), IEEE, 2023, 348–359, doi: 10.1109/CLOUD60044.2023.00048.

[15] Y. L. Li, O. Graif, U. Gupta, Toward Carbon-Efficient LLM Life Cycle, Proceedings of the 3rd Workshop on Sustainable Computer Systems (HotCarbon '24), ACM, New York, NY, USA, 2024.