

Regime-First GPU Power Management for Deep Learning: Cross-Hardware Evidence from Power, Performance, and Energy

D. Venkata Naga Raju

Vishnu Women's University

Ravi Kumar Suggala

`ravikumars@svecw.edu.in`

Vishnu Women's University

A. Sri Krishna

Vishnu Women's University

Pavan Kumar Vadrevu

Vishnu Women's University

Ratna Kanth Gudala

Vishnu Women's University

Research Article

Keywords: GPU power management, energy-efficient deep learning, power capping, thermal-aware computing, hardware characterization, sustainable computing

Posted Date: October 16th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11264835/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Regime-First GPU Power Management for Deep Learning: Cross-Hardware Evidence from Power, Performance, and Energy

D. Venkata Naga Raju¹, Ravi Kumar Suggala^{1*}, A. Sri Krishna²,
Pavan Kumar Vadrevu¹, Ratna Kanth Gudala¹

¹Department of Information Technology, Vishnu Women’s University,
Bhimavaram, Andhra Pradesh, India.

²Department of Artificial Intelligence, Vishnu Women’s University,
Bhimavaram, Andhra Pradesh, India.

*Corresponding author(s). E-mail(s): ravikumars@svecw.edu.in;

Abstract

GPU power capping is commonly used to reduce instantaneous power, but its effect on completed-work energy can depend on the hardware/workload operating regime. We characterize five deep-learning architectures on NVIDIA Tesla T4 and Quadro GV100 GPUs, evaluate hardware-aware large-language-model power predictions, validate live power-cap control on the T4, conduct a 45-run randomized complete-block T4 experiment, and analyze an additional five-block NVIDIA L4 cross-hardware validation for two completed architectures. The T4 operates in a narrow software-power-cap-dominated envelope, whereas the GV100 exhibits architecture-differentiated power and smooth thermal convergence. On the T4, reducing the cap to 60 W lowers board power but reduces throughput by 15.7–20.3% and increases energy per completed step by 6.4–11.7%. In contrast, on the L4, 60/64 W caps reduce energy per completed step by approximately 5.1–7.0% for MobileNetV2 and DenseNet-121 despite throughput losses. Hardware metadata also improves source-level power prediction, while sparse calibration transfers inconsistently across regimes. These results show that lower board power is not a reliable proxy for lower completed-work energy. We therefore propose a regime-first decision principle: empirically characterize the target GPU/workload pair and its power-performance-energy frontier before engaging external control.

Keywords: GPU power management, energy-efficient deep learning, power capping, thermal-aware computing, hardware characterization, sustainable computing

1 Introduction

GPU power and temperature management can affect throughput, energy efficiency, and execution predictability in deep-learning workloads. However, an external controller can only help if the target GPU/workload pair exposes an operating margin that native hardware management is not already exploiting. A workload far from a thermal or power constraint requires no intervention; a workload continuously constrained by a board-power envelope may offer little room for a temperature-triggered policy to improve performance or energy. This motivates a regime-first view: identify the active hardware constraint before choosing the control mechanism.

We began from a more ambitious premise: source code could be read by a large language model (LLM) to predict a workload's hardware footprint, and those predictions could be used to intervene proactively before undesirable temperature behavior emerged. A rigorous evaluation exposed a hidden assumption in that pipeline: a numerically better prediction is useful for control only if the target hardware/workload pair actually exposes controllable margin. The paper therefore asks one unifying systems question: can source-level predictions safely determine GPU power-management actions without first identifying the target hardware's realized operating regime? We answer this by linking cross-GPU characterization, prediction ablations, live intervention, and a randomized power-cap response experiment.

Characterized GPUs expose distinct operating regimes. Profiling ResNet-50, ResNet-18, MobileNetV2, ViT-B/16, and DenseNet-121 on a Tesla T4 and Quadro GV100 shows that the cards behave differently. The GV100 exhibits architecture-dependent average power and smooth RC-like temperature convergence. The T4 instead operates near a narrow board-power envelope across architectures. Follow-up live-control experiments add a crucial mechanistic observation: NVML reports software power-cap enforcement throughout sustained T4 execution, while explicit thermal-slowdown events remain absent even when temperature reaches approximately 80–82°C. Thus, the T4 is more accurately described as power-cap dominated than as a GPU with a universal 81°C thermal-throttle ceiling. An additional NVIDIA L4 power-cap experiment is used later as a cross-hardware validation of whether the T4 power-performance-energy response transfers to a newer platform.

The prediction and control results are conditional on hardware regime. In the revised LLM experiment, supplying explicit hardware metadata reduces power-prediction error on both characterized GPUs, while one-shot calibration helps the power-capped T4 but harms transfer on the architecture-differentiated GV100. Live T4 experiments show that an external temperature-triggered 66↔70 W controller can be ineffective even when it intervenes. A randomized complete-block T4 experiment at 60, 64, and 70 W demonstrates why a simple 'lower power is better' rule is unsafe: 60 W consistently reduces throughput enough to increase energy per completed step, while 64 W lies near a workload-dependent transition region rather than constituting a universal optimum. The L4 validation then shows that this energy penalty is not hardware invariant: for the two fully completed architectures, lower caps reduce energy per step even while throughput decreases.

Figure 1 summarizes this regime-first workflow. Rather than moving directly from prediction to control, the workflow uses measured telemetry and a short empirical

probe to identify the operating regime first. The resulting evidence then determines whether the workload should retain native management or whether an external power-management action is worth evaluating.

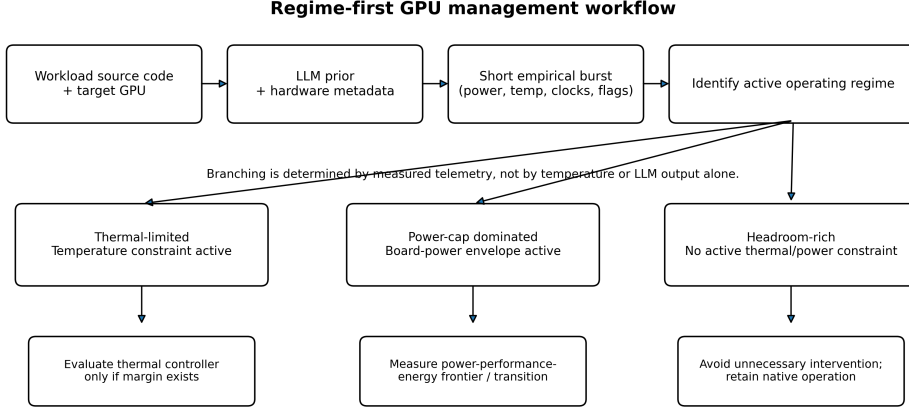


Fig. 1 Regime-first GPU management workflow for identifying the operating regime before power-control decisions.

Contributions. The primary contribution is cross-hardware evidence that the measured energy response to GPU power-cap reduction is not portable across the tested platforms and can reverse sign: a lower cap reduces board power yet increases energy per completed training step in the tested T4 cases, while reducing completed-step energy in the two completed L4 cases. The supporting systems contribution is a regime-first decision framework that uses direct telemetry and a short empirical probe to identify the active constraint and quantify control authority before selecting a controller or cap. Three secondary results support this central claim: (1) five-architecture characterization separates a narrow software-power-cap-dominated T4 envelope from architecture-differentiated GV100 behavior; (2) hardware-aware LLM prediction improves with explicit device context, but sparse calibration transfers differently across regimes and is therefore treated only as a prior; and (3) live controller experiments show that an external temperature-triggered policy adds little value in the tested T4 conditions when the measured operating regime exposes little useful authority. These observations establish non-portability across the tested platforms; they do not establish architecture-wide behavior for all T4 or L4 workloads.

Study questions. We organize the evaluation around three research questions: RQ1—what operating constraint is active for each characterized GPU/workload pair, and what telemetry distinguishes power-cap-dominated, thermally constrained, and headroom-rich behavior? RQ2—does a reduction in the power limit preserve the sign of its completed-step energy effect across tested hardware, or can that response reverse with operating regime? RQ3—once the regime is identified, how useful are source-level

LLM predictions, sparse calibration, and external control as decision aids rather than assumed control signals?

2 Related Work

2.1 Thermal-Aware and DVFS-Based Scheduling

Thermal- and power-aware scheduling is already well established, so our contribution is not a new claim that temperature or power should influence scheduling. TAPAS, for example, uses historical power/temperature information for placement, routing, and reconfiguration of LLM inference workloads at cluster scale [2]. DEPO performs online dynamic GPU power capping for CNN training and reports substantial energy savings on A100 and RTX A4500 platforms [16]. PctoDL goes further for multi-tenant inference by combining spatial partitioning, batch size, coordinated frequency control, a thermodynamic model-predictive controller, and online refinement on RTX 3080 Ti and A100 GPUs [13]. These systems demonstrate that sophisticated control can be valuable when exploitable control margin exists. Our complementary question precedes controller design: how should a runtime establish whether the target GPU/workload pair is power-cap dominated, thermal-limited, or headroom-rich, and therefore whether an additional controller has useful authority at all?

Power capping itself is likewise not novel. Multi-GPU DNN-training studies show that selected non-default caps can improve energy and composite performance-energy metrics [12], while DEPO dynamically searches caps online [16]. More recently, an offline-online DNN optimization framework jointly tunes power limits, frequency configuration, and batch size using KNN-XGBoost prediction plus online correction, reporting average reductions in both GPU energy and runtime [17]. Contemporary work on modern A40/A100/H100/H200 systems further shows workload- and architecture-dependent transitions between efficient and inefficient energy regimes under frequency and power-cap changes [18]. These results reinforce rather than subsume our question: we test whether the direction of the completed-work energy response to a qualitatively similar power-cap intervention can reverse across GPU platforms, and connect that response to measured control authority and operating-regime telemetry. Accordingly, we measure energy per completed unit of work and do not infer energy efficiency from lower instantaneous board power.

2.2 Data-Driven and Physics-Informed GPU Power/Performance Prediction

A substantial body of work predicts GPU power or performance from structured workload features and empirical data. PowerTrain uses transfer learning plus limited profiling to predict DNN training time and power on accelerated edge devices [4]. WattGPU uses public LLM and GPU metadata to predict inference power and latency on unseen server GPUs and LLMs without target profiling [5]. Physics-aware forecasting combines thermal RC structure with learned time-series models for AI-datacenter power prediction [6], while recent MFU-based work reports highly accurate

power models when conditioning on GPU, precision, and batch size [7]. Forecasting-based energy management similarly builds a perception layer from high-frequency operational telemetry [15]. These studies show that cross-hardware prediction is possible; our results should not be read as an argument to the contrary. We instead test a weaker-information setting - general-purpose source-level LLM reasoning - and ask whether metadata and a single calibration observation become reliable enough to justify control across qualitatively different operating regimes.

2.3 LLMs for Source-Level Performance Prediction

Recent work has begun testing LLMs directly as performance reasoners. Bolet et al. ask whether an LLM can classify GPU kernels as compute- or bandwidth-bound from source and target-hardware information; reasoning-capable models reach only 64% zero-shot accuracy, while explicit profiling information makes the task much easier [8]. LLMPeef studies LLM-based execution-time estimation for OpenCL programs and reports substantial error on public programs, illustrating the difficulty of source-only numerical performance modeling [9]. Our study complements these efforts by evaluating continuous board-power prediction for complete DNN training workloads across two GPUs with qualitatively different operating regimes. Rather than reporting only aggregate error, we test whether predictions preserve the hardware's cross-workload structure and whether a single empirical calibration point transfers across architectures.

2.4 Positioning of the Present Work

The closest literature establishes strong individual ingredients: dynamic power capping can save energy [16]; structured prediction and transfer can estimate performance/power [4], [5]; thermodynamic MPC and joint resource/frequency optimization can improve throughput under strict power constraints [13]; offline-online prediction can guide DNN energy optimization [17]; and recent analytical work identifies workload- and architecture-dependent GPU energy-efficiency regime shifts [18]. Our novelty claim is therefore deliberately narrower. We combine regime identification with direct throttle-reason telemetry, prediction/calibration tests, live intervention, and randomized power-cap experiments, and show that the sign of the measured energy-per-step response differs between the tested T4 and completed L4 cases. The experimental chain is source-level LLM prior -> explicit hardware context -> sparse calibration -> cross-GPU regime identification -> live intervention -> measured power-performance-energy response. A no-benefit controller result and a failed calibration transfer are retained as supporting negative evidence. This distinguishes our regime-first diagnostic question from downstream optimization.

3 Methodology

A. Regime-First Decision Framework

The framework operationalizes the paper's central claim as a gate before optimization. For a target GPU/workload pair, the controller first establishes the active

constraint from measured board power, configured power limit, temperature trajectory, clock behavior, utilization, and NVML throttle-reason telemetry. It then performs a short candidate-cap probe and evaluates realized power, throughput, and energy per completed training step. Control is enabled only when the probe reveals useful authority: a lower cap or external intervention must change the objective in a favorable direction rather than merely reduce instantaneous watts. Algorithm 1 is a decision procedure derived from the experimental evidence, not a universally validated regime classifier; its regime labels are assigned from the observed telemetry and active constraints rather than from fixed temperature or wattage thresholds.

Algorithm 1 — Regime-first control gate

- 1) Observe: collect synchronized power, temperature, clocks, utilization, configured limit, and throttle-reason telemetry during a short steady workload burst.
- 2) Diagnose: classify the observed state as power-cap dominated, thermally constrained, or headroom-rich from the active constraint and trajectory; do not infer the regime from temperature alone.
- 3) Probe: test one or more safe candidate limits and measure realized power P , throughput S , and completed-step energy $E_{\text{step}} = P/S$ under matched run preparation.
- 4) Quantify authority: compare the probe with native operation. A power reduction without a favorable completed-work energy/performance response is not treated as useful authority.
- 5) Gate and select: if useful authority exists, choose the simplest policy supported by the measured response (e.g., static cap or dynamic control); otherwise retain native management. LLM output, when available, can prioritize probes but cannot bypass empirical gating.

B. Experimental Hardware and Model Architectures

We evaluate on two GPUs with substantially different power envelopes: an NVIDIA Tesla T4 (70W TDP, inference-oriented datacenter card) and an NVIDIA Quadro GV100 (250W TDP, workstation-class card). This pairing was chosen deliberately after preliminary profiling revealed that the two cards occupy qualitatively different operating regimes (Section 4.1), making single-GPU claims about thermal-aware scheduling non-generalizable.

Five convolutional and transformer architectures of varying compute character were used throughout: ResNet-50, ResNet-18, MobileNetV2, ViT-B/16, and DenseNet-121, each trained with batch size 64 at 224×224 input resolution using randomly initialized weights (network architecture and numerical compute pattern were the object of study, not converged model behavior).

C. Phase 1: GPU Thermal and Power Profiling

1) Profiling protocol:

For each (architecture, GPU) pair, we recorded GPU temperature and power draw at 10Hz via NVML while running continuous forward/backward training steps. Profiling methodology was iteratively hardened in response to several confounds discovered

during data collection, each of which we report because they materially changed measured values and would affect any similar profiling study:

- **Order confound.** An initial protocol that profiled all five architectures sequentially without inter-run cooldown produced a monotonically increasing average temperature pattern that closely tracked execution order rather than architecture identity. We corrected this by (a) inserting an explicit cooldown-to-baseline period between profiling runs and (b) randomizing model execution order independently across repeats.
- **Initial-condition confound in fixed-window deltas.** A naïve metric (temperature delta over a fixed profiling window) was found to depend heavily on the GPU's starting temperature rather than architecture, because first-order thermal heating follows ' $dT/dt \propto (T_{ss} - T_{current})$ ': identical workloads produce smaller observed deltas when started from a hotter baseline. We replaced fixed-window deltas with joint exponential curve fitting (Section 3.2.2) that is structurally invariant to starting temperature.
- **Window-duration-vs-time-constant identifiability.** A 15 s profiling window produced unstable and underdetermined curve fits for architectures with longer thermal time constants. Profiling duration was therefore extended to 150 s so that the final window comfortably covered the slowest observed heating dynamics (T4 rise descriptors up to approximately 53 s), improving identifiability and reducing dependence on the arbitrary endpoint of a short trace.
- **Hidden warm-up transient.** An initial protocol excluded 5 "warm-up" training steps from the recorded trace (intended to exclude one-time CUDA/cuDNN kernel-selection overhead). Because warm-up wall-clock time is architecture-dependent (proportional to per-step compute time), this silently truncated a different fraction of each architecture's initial heating transient, systematically compressing fitted time constants for fast-stepping architectures. The protocol was corrected to begin recording before warm-up, retaining only 2 lightweight throwaway steps for kernel selection.
- **Non-monotonic T4 traces.** Several T4 profiling repeats rose, peaked near 80–81°C, and then declined before the window ended, a shape that a monotonic single-exponential model cannot represent reliably. We therefore treated these traces as non-monotonic operating-regime observations rather than forcing a thermal-RC fit or inferring a hardware thermal-throttle threshold from temperature alone. Subsequent live NVML telemetry was used to identify the active constraint directly: software power-cap enforcement was persistent, while software/hardware thermal-slowdown flags were not observed. For descriptive cold-start analysis, a smoothed peak-and-settle detector was retained to summarize rise time without assigning the peak a causal throttle interpretation.

2) *GV100 thermal model: joint RC curve fitting:*

For the GV100, repeated profiling across all five architectures showed smooth exponential convergence and no observed NVML thermal-slowdown behavior in the

tested runs. We therefore modeled each architecture's thermal trace as a first-order RC system,

$$T(t) = T_{ss} - (T_{ss} - T_0) \cdot \exp(-t/\tau),$$

fitting T_{ss} and τ jointly across repeats with heterogeneous starting temperatures T_0 (fixed per repeat, not fitted). This formulation reduces confounding between starting temperature, steady-state temperature, and time constant; synthetic reconstruction checks were used to confirm that joint fitting was more stable than fitting short repeats independently.

We further calibrated the RC model's physical constants (thermal resistance R , cooling coefficient k) via linear regression of fitted T_{ss} against measured average power across the five architectures: $T_{ss} = T_{amb} + (R/k) \cdot P$. This regression achieved $R^2 = 0.995$, and the independently-fitted cooling coefficient $k = 1/\tau$ varied by only 5.3% (coefficient of variation) across the five architectures — consistent with k being a property of the physical card rather than the workload, which is the assumption the single-RC-constant model requires and which we verified rather than assumed.

3) *T4 operating-regime characterization:*

The T4 requires a different interpretation from the GV100. Cold-start traces often rise toward approximately 80–81°C and later settle near 76–77°C, but live NVML throttle-reason telemetry collected during subsequent real power-control experiments shows persistent `nvmlClocksThrottleReasonSwPowerCap` activity and no observed software/hardware thermal-slowdown flags. We therefore avoid treating 81°C as a verified hardware thermal-throttle threshold. Instead, we characterize the T4 as a power-cap-dominated regime in which architecture-specific workloads operate close to the board-power envelope and exhibit temperature-dependent clock behavior within that constraint.

For descriptive analysis we retain workload-specific rise-time statistics from cold-start profiling, because they quantify how quickly different architectures heat the card. However, these parameters are no longer interpreted as time-to-thermal-throttle. In the live-control phases, the directly observed variables—power limit, board power, SM clock, temperature, utilization, and NVML throttle-reason bitmasks—are treated as the authoritative indicators of the active hardware constraint.

D. Phase 2: LLM-Based Hardware Footprint Prediction

We evaluated gpt-oss-120b via the Groq API under three prompting conditions. Z1 supplied the exact FP32 training workload specification and target GPU name; Z2 additionally supplied public hardware metadata; Z3 supplied the same metadata plus one empirical ResNet-18 average-power calibration point and evaluated transfer to the remaining four architectures. The model directly predicted time-averaged board power per training step, avoiding the physically invalid practice of averaging independently predicted forward- and backward-pass powers without duration weighting. Both temperature 0.0 and 0.2 were tested in the pilot, and raw responses, provenance, validation status, and repeated inference replicates were recorded. Prediction error was scored against measured Phase-1 power using MAE/MAPE, while hardware-regime awareness was assessed from the predicted cross-architecture power spread.

E. Phase 3: Live Power-Cap, Controller, and Knee Evaluation

1) *Establishing the real control surface:*

Before evaluating a controller, we verified that the Colab-hosted Tesla T4 allowed live NVML power-limit writes over its supported 60–70 W range. An initial screening sweep across 60, 62, 64, 66, 68, and 70 W established the real control surface and showed that progressively lower caps reduce SM clock and throughput. We recorded board power, temperature, utilization, clock frequency, throttle-reason bitmasks, step time, throughput, and energy per completed step at approximately 10 Hz.

2) *Live controller validation:*

We compared native 70 W operation, static 66 W operation, and a hysteretic 66↔70 W temperature-triggered policy. Runs were prepared to a common thermal band before measurement and policy order was randomized within blocks. The live controller test was used to determine whether an external temperature-based intervention adds value when native T4 power management is already active; ResNet-18 provided valid interventions, while ViT-B/16 supplied a complementary non-intervention case under the tested trigger.

3) *Randomized power-cap knee experiment:*

To test whether a simpler static operating point could outperform or match dynamic intervention, we conducted a randomized complete-block experiment on MobileNetV2, DenseNet-121, and ResNet-50 at 60, 64, and native 70 W. Each architecture used five blocks, with cap order randomized within each block, 60 s measured runs, FP32 batch size 64 at 224×224 resolution, and thermal-state preparation targeting 76–77°C before each run. Primary outcomes were throughput (steps/s), energy per completed step, realized board power, temperature, and SM clock. The experimental unit for inference was the run/block, not individual telemetry samples. We report run-level means with 95% confidence intervals and paired within-block contrasts against 70 W. Because only five independent blocks were available per architecture and multiple outcomes/caps were explored, p-values are treated as descriptive supporting evidence rather than confirmatory familywise inference; effect magnitudes and confidence intervals remain primary. The measured start temperature occasionally read 75°C immediately after preparation; this one-degree transition occurred across policies and is reported as a protocol limitation rather than silently excluded.

4) *Experimental provenance and reproducibility.* Every live run records the physical GPU UUID, configured power limit, randomized block order, random seed, timestamps, run-level summary, step durations, and raw NVML telemetry. Checkpoints were written after each completed run, and a resumed session was allowed to append only when the GPU UUID matched; otherwise a new dataset was created. This prevents silent mixing of cloud GPU instances within a controlled comparison and preserves the raw evidence used for all reported block-level analyses. The run-level JSON, derived analysis CSVs, and experiment scripts are retained as reproducibility artifacts and can accompany the manuscript as supplementary material.

4 Results

4.1 Cross-Hardware Operating-Regime Characterization

As shown in Table 1, the GV100 exhibits architecture-dependent thermal and power behavior, while the fitted thermal time constants remain relatively consistent across

the tested architectures. *ResNet-50 did not reach the approximately 81°C peak

Table 1 GV100 fitted thermal parameters.

Architecture	T_{ss} (°C)	τ (s)	Avg. Power (W)
ResNet-50	74.17 ± 0.15	14.16 ± 0.21	201.09
ResNet-18	73.14 ± 0.22	15.69 ± 0.31	192.63
MobileNetV2	69.30 ± 0.13	13.44 ± 0.21	168.68
ViT-B/16	77.60 ± 0.13	15.04 ± 0.17	220.07
DenseNet-121	71.28 ± 0.13	14.30 ± 0.21	179.18

Calibrated RC constants: $R = 0.01082$, $k = 0.0688$ (CV 5.3% across architectures), $T_{amb} = 42.88^\circ\text{C}$, T_{ss} -vs-power fit $R^2 = 0.995$.

Table 2 T4 cold-start heating characteristics and average power.

Architecture	Observed peak (°C)	Time-to-peak (s)	τ_{rise} (s)	Post-peak settle (°C)	Avg. Power (W)
ResNet-18	81.0	84.2	36.31	77.0	67.30
MobileNetV2	81.0	92.7	37.78	76.5	66.65
ViT-B/16	81.0	113.6	45.76	76.0	66.63
DenseNet-121	80.9	127.8	52.64	76.9	67.17
ResNet-50	78.0*	> 150 (not reached)	51.29	78.0*	67.58

within the 150 s profiling window. As shown in Table 2, the T4 workloads exhibit similar average power despite differences in their cold-start heating behavior. Most workloads approach an observed peak near 81°C, while their time-to-peak and rise characteristics vary across architectures. Two findings emerge. First, average board power is nearly uniform across architectures on the T4 (66.63–67.58 W, about 1.4% relative spread) but substantially differentiated on the GV100 (168.68–220.07 W, about 26% spread). Second, the live T4 experiments identify the mechanism behind this uniformity more directly than the temperature trace alone: software power-cap enforcement is active throughout sustained execution, while explicit NVML thermal-slowdown events are absent. The relevant contrast is therefore a power-cap-dominated T4 versus a GV100 that preserves architecture-dependent power differentiation and shows no observed thermal slowdown in the tested runs.

As shown in Fig. 2, the GV100 workloads exhibit smooth RC-like convergence with architecture-dependent power, while the T4 workloads operate near a narrow board-power envelope. The approximately 80–81°C T4 peak is treated as an observed temperature feature rather than a verified thermal-throttle threshold.

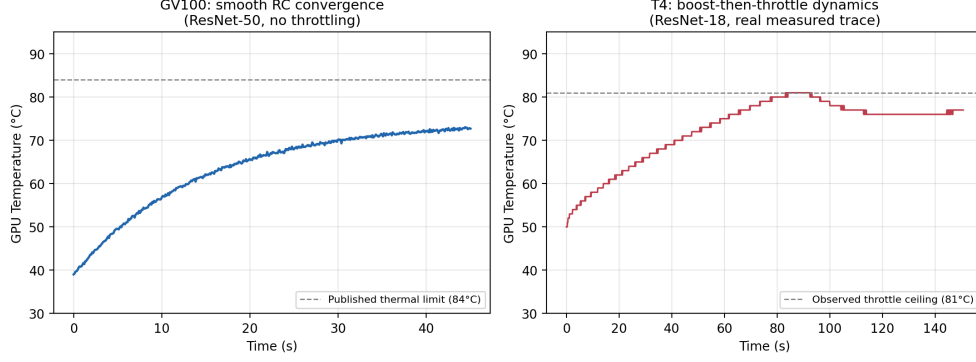


Fig. 2 Measured thermal dynamics across hardware regimes.

4.2 LLM Prediction: Hardware Context Helps, Calibration Is Regime-Dependent

Table 3 Power-prediction MAPE across prompting conditions.

GPU	Z1 MAPE	Z2 MAPE	Z3 MAPE	Metadata	1-shot calibra- tion	Evaluation scope
Tesla T4	21.49%	13.70%	9.81%	Improves	Improves further	4 non-calibration architectures
Quadro GV100	14.38%	11.35%	22.60%	Improves	Degrades	4 non-calibration architectures
Z1				GPU name only		
Z2				GPU name + public hard- ware metadata		
Z3				Metadata + one ResNet- 18 measured power value		
Key result				Hardware con- text helps; cal- ibration trans- fer is regime- dependent		

As shown in Table 3, hardware metadata improves prediction accuracy on both GPUs, while one-shot calibration has different effects across the two operating regimes.

Explicit hardware context consistently helps: moving from GPU-name-only prompting (Z1) to hardware metadata (Z2) reduces MAPE from 21.49% to 13.70% on the T4 and from 14.38% to 11.35% on the GV100. One-shot calibration (Z3) further reduces T4 MAPE to 9.81%, but increases GV100 MAPE to 22.60%. This asymmetry is mechanistically consistent with the measured regimes: one calibration point is

informative when the T4 collapses workloads toward a common power envelope, but can be misleading on the GV100 where architecture-specific power differences are real.

Table 4 Qualitative reliability findings from the LLM experiment.

Finding	T4	GV100
Power regime in measurements	~1.4% spread	architecture ~26% architecture spread
Z1 zero-shot prediction	Over-differentiates workloads	Qualitatively differentiated
Hardware metadata (Z2)	Reduces MAPE	Reduces MAPE
One-shot calibration (Z3)	Further reduces MAPE	Increases MAPE
Control implication	Calibration can reveal shared envelope	Single-point transfer can mislead

As summarized in Table 4, the qualitative reliability findings further highlight the hardware-dependent behavior of prediction and calibration across the T4 and GV100. These results do not support using a zero-shot LLM estimate as a direct control signal. The model benefits from hardware metadata and can sometimes exploit sparse calibration, but the value of that calibration itself depends on the hardware regime. A practical system should therefore treat the LLM output as a prior and update it with a short hardware measurement rather than assume that semantic workload knowledge uniquely determines realized board power or thermal behavior.

As shown in Fig. 3, explicit device metadata reduces prediction error on both GPUs, while one-shot calibration improves the power-capped T4 but degrades the architecture-differentiated GV100, showing that calibration transfer is regime-dependent.

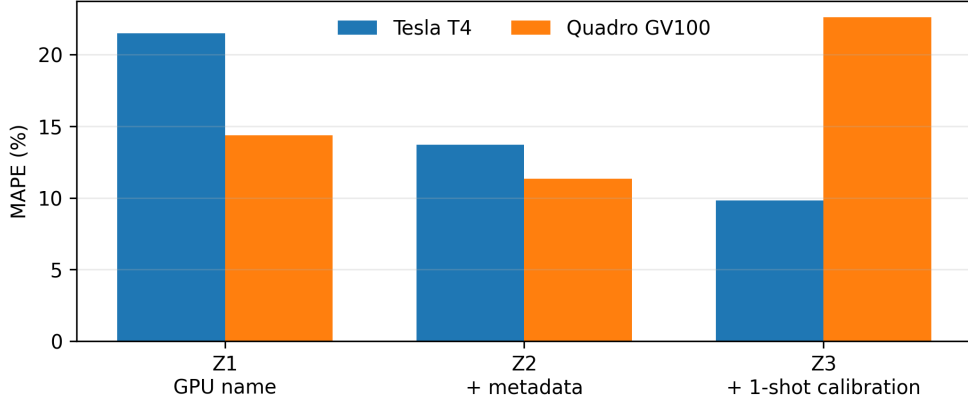


Fig. 3 LLM power prediction across hardware regimes.

4.3 Live T4 Control and Power-Cap Transition Results

Screening and controller validation. The initial sweep established a monotonic control surface: lowering the T4 cap reduces achievable clock and throughput, and lower instantaneous board power does not guarantee lower energy per completed step. In the live ResNet-18 controller experiment, valid 66↔70 W interventions produced essentially unchanged throughput, energy per step, mean temperature, peak temperature, and hot-state exposure relative to native 70 W operation. For ViT-B/16, the dynamic policy did not trigger under the tested threshold. These negative results motivated a direct test of static operating points rather than further tuning of the temperature controller.

As summarized in Table 5, the ResNet-18 values aggregate completed valid interventions from the live 66↔70 W controller experiment, while the ViT-B/16 values come from the three-block validation. Dynamic ViT runs are reported as non-interventions because the 79°C trigger was never reached.

Run-level dispersion: ResNet-18 native throughput 4.7104 ± 0.0026 steps/s and energy 13.878 ± 0.029 J/step ($n=5$); four valid dynamic interventions 4.7107 ± 0.0044 steps/s and 13.859 ± 0.040 J/step ($n=4$). ViT-B/16 native throughput 0.4381 ± 0.0017 steps/s and energy 147.73 ± 0.63 J/step ($n=3$); dynamic non-intervention runs 0.4364 ± 0.0018 steps/s and 148.53 ± 0.54 J/step ($n=3$). Values are mean $\pm$ sample SD across runs/blocks.

As shown in Fig. 4, lowering the board-power cap reduces achievable clock/throughput and does not guarantee lower energy per completed step. In valid ResNet-18 dynamic runs, the 66–70 W controller changes the commanded cap but produces essentially no change in mean temperature, hot-state exposure, throughput, or energy relative to native 70 W operation.

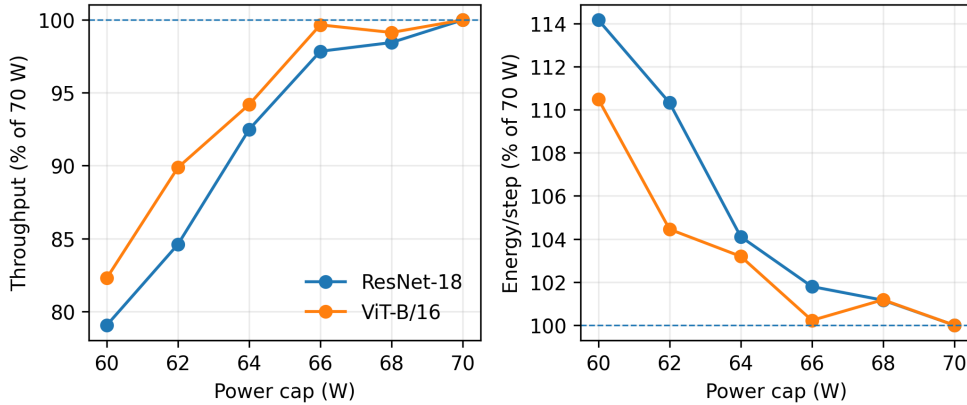


Fig. 4 Live T4 power-control results.

Randomized-block power-cap experiment. The 45-run follow-up provides the strongest live evidence in the paper. Relative to native 70 W, a 60 W cap reduced realized board power by 10.35% for MobileNetV2, 10.44% for DenseNet-121, and 10.96%

Table 5 Live policy comparison on the T4.

Architecture / policy	Throughput (steps/s)	Energy / step (J)	Mean temp (°C)	Peak temp (°C)	Time $\geq 80^\circ\text{C}$
ResNet-18 / native 70 W	4.7104	13.878	77.542	82.0	26.52%
ResNet-18 / static 66 W	4.6953	13.936	77.546	82.0	26.82%
ResNet-18 / dynamic 66 $\leftrightarrow$ 70 W	4.7107	13.859	77.538	82.0	26.79%
ViT-B/16 / native 70 W	0.4381	147.730	76.925	77.0	0.00%
ViT-B/16 / static 66 W	0.4338	149.030	77.472	79.33	12.51%

for ResNet-50, but throughput fell by 15.72%, 17.33%, and 20.28%, respectively. Consequently, energy per completed step increased by 6.37%, 8.32%, and 11.70%. The 64 W condition was qualitatively different: realized board power fell by about 4.65–5.12%, while throughput penalties were 4.78%, 5.45%, and 8.49%; energy per step increased by only 0.13% for MobileNetV2, 0.81% for DenseNet-121, and 3.68% for ResNet-50. We therefore interpret 64 W as a workload-dependent transition region on this T4, not as a universal optimum or a formally estimated knee.

As summarized in Table 6, the randomized complete-block T4 experiment uses five blocks per architecture and cap. Values are run-level means, and percentage changes are paired within block against native 70 W.

Table 6 Randomized complete-block T4 power-cap experiment.

Architecture	Cap	Throughput (steps/s)	Energy/step (J)	Power (W)	Δ Throughput vs 70 W	Δ Energy/step vs 70 W
MobileNetV2	60 W	3.673	16.31	59.96	-15.72%	+6.37%
MobileNetV2	64 W	4.150	15.36	63.76	-4.78%	+0.13%
MobileNetV2	70 W	4.358	15.34	66.88	Reference	Reference
DenseNet-121	60 W	1.332	45.02	60.02	-17.33%	+8.32%
DenseNet-121	64 W	1.524	41.90	63.90	-5.45%	+0.81%
DenseNet-121	70 W	1.611	41.57	67.02	Reference	Reference
ResNet-50	60 W	1.318	45.43	59.87	-20.28%	+11.70%
ResNet-50	64 W	1.512	42.17	63.80	-8.49%	+3.68%
ResNet-50	70 W	1.653	40.68	67.25	Reference	Reference

As shown in Fig. 5, lowering the cap reduces realized board power, but the throughput penalty is nonlinear and architecture-dependent. The 64 W setting lies in a milder transition region, whereas 60 W is clearly beyond that efficient transition for all three tested workloads.

The energy view is essential to interpreting this transition. As shown in Fig. 6, the 60 W setting increases joules per completed step for all three architectures despite reducing instantaneous board power.

Fig. 7 confirms that the configured caps do reduce realized board power; therefore, the energy penalty is explained by the disproportionate throughput loss rather than by failure of the cap to take effect.

As summarized in Table 7, the paired block effects are reported relative to native 70 W as mean percentage changes with 95% confidence intervals across five paired blocks. Negative throughput indicates a performance loss, while positive energy/step indicates worse energy efficiency.

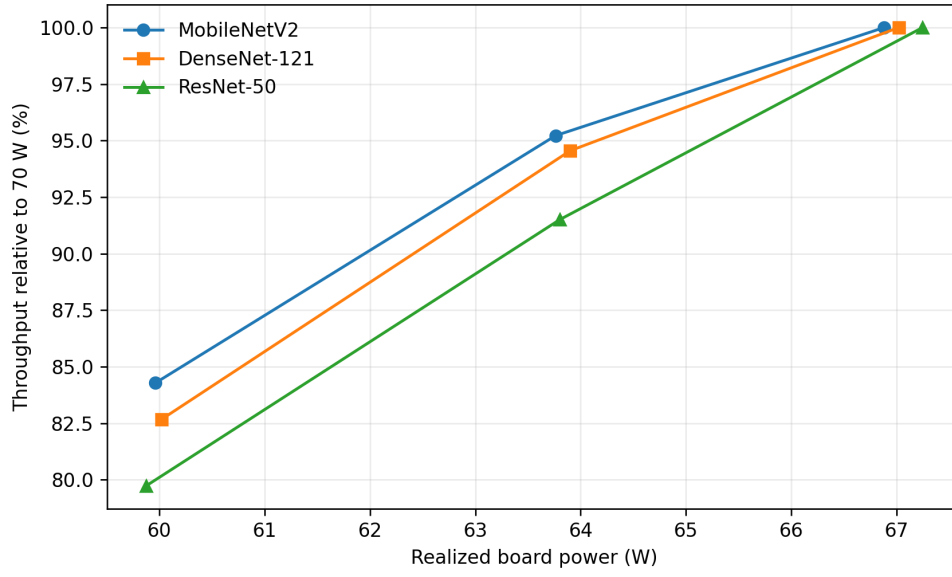


Fig. 5 Measured T4 power-performance transition.

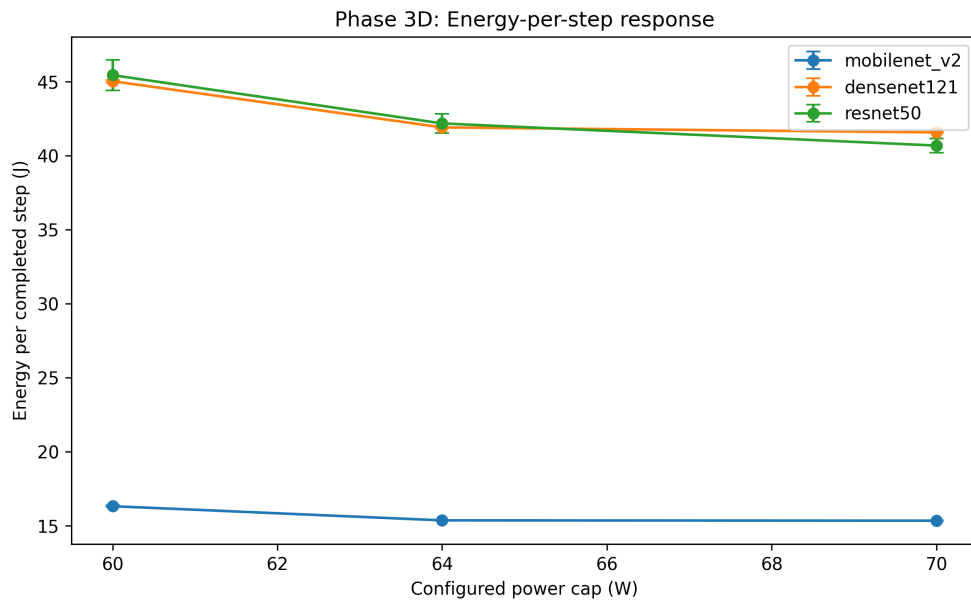


Fig. 6 Energy per completed training step on the T4.

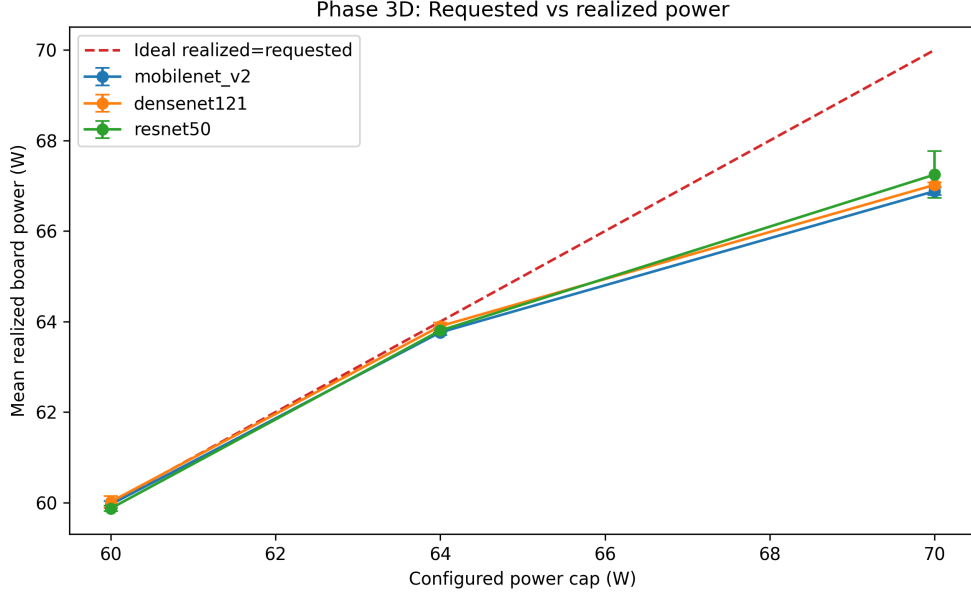


Fig. 7 Realized T4 board-power response.

Table 7 Paired block effects relative to native 70 W.

Architecture	Cap	Δ Throughput	95% CI	Δ Energy/step	95% CI
MobileNetV2	60 W	-15.72%	[-16.01, -15.43]%	+6.37%	[+6.01, +6.73]%
MobileNetV2	64 W	-4.78%	[-4.89, -4.67]%	+0.13%	[+0.005, +0.258]%
DenseNet-121	60 W	-17.33%	[-17.49, -17.17]%	+8.32%	[+8.04, +8.60]%
DenseNet-121	64 W	-5.45%	[-5.56, -5.33]%	+0.81%	[+0.64, +0.98]%
ResNet-50	60 W	-20.28%	[-21.60, -18.95]%	+11.70%	[+9.42, +13.98]%
ResNet-50	64 W	-8.49%	[-9.76, -7.22]%	+3.68%	[+1.91, +5.45]%

4.4 Cross-Hardware Power-Cap Validation on NVIDIA L4

To test whether the T4 cap-response direction transfers across hardware, we analyzed an additional NVIDIA L4 experiment using the same paired randomized-block logic. MobileNetV2 and DenseNet-121 each completed five blocks at 60, 64, and native 72 W. Relative to 72 W, MobileNetV2 throughput decreased by 11.96% at 60 W and 5.95% at 64 W, while energy per completed step decreased by 5.05% and 5.38%, respectively. DenseNet-121 throughput decreased by 11.23% at 60 W and 5.55% at 64 W, while energy per step decreased by 7.04% and 6.04%. Thus, unlike the T4—where lower caps increased joules per completed step—the L4 converts the power reduction into a net energy saving for both fully completed architectures, although at a throughput cost.

This cross-hardware reversal provides direct evidence that the response to a cap reduction is not portable across the tested platforms: the sign of the completed-step

energy response cannot be inferred from the cap reduction alone, and an absolute setting such as 64 W cannot be treated as a portable optimum. The reversal coincides with—and is mechanistically consistent with—the substantially different measured operating regimes and telemetry of the tested platforms; because the experiments do not isolate GPU architecture from every cloud-instance and software-environment difference, we do not attribute the reversal to GPU architecture alone. Cross-hardware validation on the NVIDIA L4 was completed only for the architectures reported below; missing ResNet-50 L4 observations are neither imputed nor generalized.

As shown in Fig. 8, the T4 points move toward higher energy per completed step as caps are reduced, whereas the completed L4 points move toward lower energy per step despite throughput loss. Absolute wattages are platform-specific and are not interpreted as portable optima.

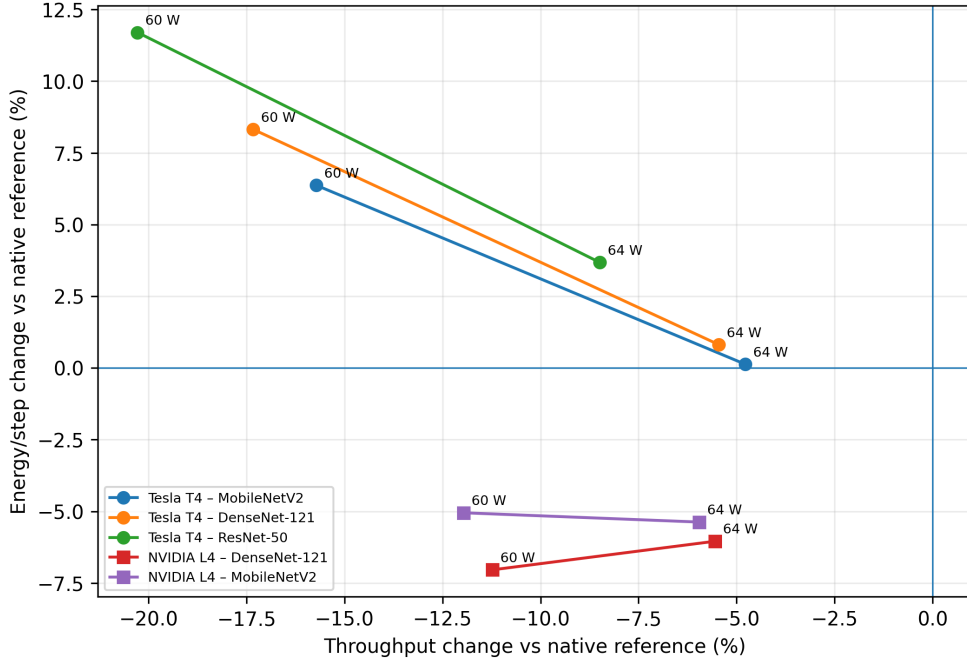


Fig. 8 Cross-GPU energy-throughput response.

Table 8 reports the paired L4 energy-per-step comparisons against native 72 W across five blocks.

E. Summary

The combined evidence supports the central claim of the paper. The T4 does not support a universal fixed-temperature thermal-throttle interpretation; direct telemetry instead identifies persistent software power-cap enforcement in the tested sustained runs. External temperature-triggered control offers no meaningful advantage in the live cases tested, while the randomized static-cap experiment reveals a nonlinear,

Table 8 L4 paired energy-per-step comparisons against native 72 W.

Architecture	Cap	Mean J/step	Δ vs 72 W	Paired Δ (J)	95% CI (J)	p
DenseNet-121	60 W	11.769	-7.04%	-0.891	[-1.040, -0.743]	7.60e-05
DenseNet-121	64 W	11.895	-6.04%	-0.765	[-0.802, -0.728]	5.43e-07
MobileNetV2	60 W	4.984	-5.05%	-0.265	[-0.278, -0.253]	5.13e-07
MobileNetV2	64 W	4.967	-5.38%	-0.282	[-0.298, -0.266]	9.68e-07

architecture-dependent power-performance-energy transition. A 60 W cap consistently saves instantaneous board power but increases energy per completed step on the tested T4 workloads. Crucially, the additional L4 validation reverses the energy direction for the two fully completed architectures: 60/64 W caps reduce energy per step while reducing throughput. Together with the architecture-differentiated GV100 characterization, these observations show that power-management actions should be conditioned on the measured hardware/workload regime rather than transferred as fixed-wattage rules.

5 Discussion

5.1 Thermal-aware control should be gated by operating regime

The principal systems implication is that thermal-aware control should be gated by operating regime rather than enabled by default. A runtime should first determine whether the target workload is genuinely approaching a thermal constraint, is instead constrained by board power, or has substantial thermal headroom. This regime check can use a short telemetry burst that jointly observes temperature trajectory, board power, clocks, utilization, and NVML throttle-reason flags. Only the first case directly motivates a temperature-triggered controller. This interpretation is consistent with recent production-scale thermal/power scheduling systems, which rely on measured operating context rather than assuming a universal thermal bottleneck [2], and with recent power-constrained GPU optimization showing that the best control depends on the available power/performance surface [13].

5.2 Lower power is not equivalent to lower energy

The randomized T4 experiment makes this point quantitatively. At 60 W, realized board power falls by roughly 10–11% relative to native 70 W, yet throughput falls by 15.7–20.3%; the longer execution time dominates, raising energy per completed step by 6.4–11.7%. At 64 W, the trade-off is milder but remains architecture-dependent. The L4 validation demonstrates that this relationship is not universal: for MobileNetV2 and DenseNet-121, both 60 W and 64 W reduce energy per completed step relative to native 72 W even though throughput also falls. Therefore neither instantaneous power reduction nor a fixed cap value is sufficient to predict energy efficiency; the relevant quantity is completed-work energy on the target hardware/workload pair.

5.3 Native GPU control can dominate an external controller

The ResNet-18 live test is a direct negative result for the external hysteretic controller: the policy truly intervenes but does not materially change throughput, energy per step, mean temperature, peak temperature, or hot-state exposure. ViT-B/16 supplies the complementary case in which the controller does not need to intervene. The subsequent 60/64/70 W experiment explains why further temperature-controller tuning is not automatically desirable: native T4 management already places the device in a power-constrained regime, and forcing a lower cap can move a workload beyond its efficient operating region. A practical controller should therefore first establish that controllable margin exists and then optimize on an empirically measured power-performance-energy frontier.

Relation to recent optimizers. Our findings do not imply that advanced GPU controllers are ineffective. DEPO demonstrates sizable energy savings on Ampere GPUs when online cap adaptation can exploit performance-energy slack [16], and PctoDL demonstrates that thermodynamic MPC combined with partitioning and frequency control can substantially improve power-constrained multi-tenant inference [13]. The distinction is one of precondition: those systems ask how to optimize once a useful control surface exists, whereas our experiments ask whether that control surface exists for the target hardware/workload pair. On the tested T4, persistent native software-power-cap enforcement and the 60/64/70 W response show why blindly adding a temperature controller can be unnecessary or counterproductive. This motivates a regime classifier or uncertainty gate ahead of any downstream DVFS, MPC, capping, or configuration-search optimizer.

5.4 LLM predictions are useful priors, not control authority

The revised LLM ablation gives a more nuanced result than a single MAPE number. Hardware metadata improves prediction on both GPUs, showing that the model can use explicit device context. One-shot calibration improves the T4 but degrades the GV100, demonstrating that sparse calibration is not automatically transferable. This is consistent with the broader prediction literature: strong cross-hardware results typically rely on structured hardware features, empirical training corpora, or carefully conditioned models [4], [5], [7], whereas source-only LLM performance reasoning remains substantially less reliable [8], [9]. We therefore position the LLM as a hypothesis generator or prior whose numerical output must be calibrated and uncertainty-checked before it is allowed to drive control.

5.5 Threats to validity

- Cloud GPU instance variation. T4 experiments were performed on multiple Colab-provisioned Tesla T4 instances across sessions. Within each controlled comparison, policies were kept on the same GPU UUID; measurements from different UUIDs were not treated as repeated observations of one physical device. This improves ecological validity but prevents attributing small cross-session differences to workload alone.

- Synthetic workload. The experiments use randomly initialized models and synthetic 224×224 inputs to isolate architecture-level compute behavior. Real training pipelines introduce data loading, augmentation, optimizer state, mixed precision, and changing utilization patterns that may alter the operating point.
- Limited live-policy and cap scope. The full five-architecture T4/GV100 characterization is broader than the live control validation. The dynamic controller was tested on ResNet-18 and ViT-B/16, while the randomized 60/64/70 W T4 experiment used MobileNetV2, DenseNet-121, and ResNet-50. Cross-hardware L4 validation was completed for MobileNetV2 and DenseNet-121 using five paired randomized blocks at 60/64/72 W. Experimental access to the original L4 cloud instance ended before the planned ResNet-50 extension could be completed; incomplete observations are excluded and no values are imputed. Accordingly, the L4 result demonstrates non-invariance of the observed energy-response direction across hardware, but it is not used to claim architecture-wide L4 behavior or a universal cap.
- Single LLM family. The revised prediction experiment uses gpt-oss-120b through Groq. The ablation isolates the value of hardware metadata and sparse calibration for this model, but does not establish that all LLM families exhibit identical numerical error or uncertainty behavior.

6 Limitations

- The live controller changes board-power limits only within the T4-supported 60–70 W range. It does not directly manipulate clocks, voltage, fan policy, or cooling infrastructure, so the conclusions apply specifically to power-cap intervention.
- The live experiments use controlled 45–60 s windows intended to compare operating policies under matched thermal state. Longer end-to-end training runs may expose slower environmental drift, dataloader effects, checkpointing, or cooling-system dynamics. In the randomized-block experiment, thermal preparation targeted 76–77°C, but the immediately recorded start value was occasionally 75°C; because this occurred across policies, we retain all runs and disclose the one-degree transition rather than treating the telemetry samples as perfectly identical initial states.
- The LLM calibration study uses one ResNet-18 power observation as the Z3 calibration point. More principled calibration-set selection, multi-point calibration, and uncertainty-aware Bayesian updating remain open questions.
- We report full five-architecture operating-regime characterization on T4 and GV100, a complete three-architecture five-block cap experiment on the T4, and a completed two-architecture five-block cross-hardware cap validation on the L4. Five randomized blocks establish strong within-setup repeatability but remain a small number of independent experimental units and do not characterize unit-to-unit manufacturing variation. Future replication on independently provisioned L4 systems, additional architectures, and a normalized-cap GV100 sweep would test the breadth of the regime-first interpretation; these are extensions rather than prerequisites for the observations reported here.

7 Conclusion

We investigated whether source-level LLM predictions and proactive power/thermal control can improve GPU training behavior without first requiring extensive profiling. The evidence supports a regime-first conclusion rather than a universal scheduler or fixed-cap claim. The GV100 shows architecture-differentiated power and smooth RC-like convergence, while the tested Tesla T4 sustained runs are persistently software-power-cap dominated. On the T4, a live 66↔70 W controller provides no operationally meaningful benefit for ResNet-18, and the randomized 60/64/70 W experiment shows that aggressive capping can lower instantaneous power yet increase energy per completed step. Cross-hardware validation on the L4 provides the complementary result: for the two fully completed architectures, MobileNetV2 and DenseNet-121, 60/64 W caps reduce energy per completed step relative to native 72 W despite throughput losses. This establishes that the direction of the observed power-performance-energy response is not invariant across the tested platforms; it does not imply that the same L4 response holds for every architecture, nor does it isolate GPU architecture as the sole cause of the reversal. Explicit hardware metadata also improves LLM power prediction on both characterized GPUs, but one-shot calibration transfers successfully only in the shared-envelope T4 regime and degrades prediction on the architecture-differentiated GV100. A practical runtime should therefore treat source-level predictions as priors, perform a short empirical characterization of the target hardware/workload pair, measure completed-work energy and throughput under candidate operating points, and engage external control only when the resulting frontier exposes useful margin. Lower board power is not, by itself, evidence of lower energy per completed training step.

Declarations

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Data availability

The retained reproducibility materials comprise raw run-level JSON outputs, derived analysis CSVs, randomized block information, GPU provenance, retained experiment notebooks, selected figures, and analysis/verification code for the reported results. The materials have been organized as a reviewer-ready artifact bundle with a README, file manifest, provenance and limitations documentation, duplicate/redundancy audit, and SHA-256 checksums. No missing observations were reconstructed or imputed; in particular, the interrupted L4 ResNet-50 collection is retained for provenance but excluded from the reported inferential comparisons. The artifact bundle will accompany the submission as supplementary reproducibility material and/or be deposited in a persistent repository; the repository DOI/URL will be added after deposition.

Code availability

Analysis and verification code for the reported results is included in the retained reproducibility materials described above. A persistent repository DOI/URL will be added after deposition.

Author contributions

D. V. Naga Raju: Conceptualization, Methodology, Software, Investigation, Data curation, Formal analysis, Visualization, Project administration, Writing – original draft, Writing – review & editing. S. Ravi Kumar: Methodology, Validation, Formal analysis, Supervision, Writing – review & editing. A. Sri Krishna: Methodology, Validation, Formal analysis, Visualization, Writing – review & editing.

Generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this work, the authors used OpenAI ChatGPT to support manuscript organization, language and readability improvement, literature-checking assistance, and editorial review. The authors independently reviewed and verified the scientific claims, numerical results, references, interpretations, and final text, edited the content as needed, and take full responsibility for the content of the publication.

References

- [1] Y. Lee, K. G. Shin, and H. S. Chwa, "Thermal-Aware Scheduling for Integrated CPUs-GPU Platforms," *ACM Transactions on Embedded Computing Systems*, vol. 18, no. 5s, Art. 90, 2019, doi: 10.1145/3358235.
- [2] J. Stojkovic et al., "TAPAS: Thermal- and Power-Aware Scheduling for LLM Inference in Cloud Platforms," in *Proc. 30th ACM ASPLOS*, vol. 2, pp. 1266-1281, 2025, doi: 10.1145/3676641.3716025.

- [3] R. Lu and D. Wang, "A Thermal-aware Workload Scheduler for High-performance LLM Inference in Cooling-regulated Datacenters," in Proc. 4th Workshop on Sustainable Computer Systems (HotCarbon), 2025, doi: 10.1145/3757892.3757906.
- [4] S. K. Prashanthi, S. Taluri, Beautlin S, L. Karwa, and Y. Simmhan, "PowerTrain: Fast, generalizable time and power prediction models to optimize DNN training on accelerated edges," *Future Generation Computer Systems*, vol. 161, pp. 329-344, 2024, doi: 10.1016/j.future.2024.07.001.
- [5] M. Fadel Argerich, J. Fürst, and M. Patiño-Martínez, "WattGPU: Predicting Inference Power and Latency on Unseen GPUs and LLMs," arXiv:2607.02391, 2026.
- [6] M. S. Ali AlShaikh Saleh, S. Chawla, S. Bayhan, H. Abu-Rub, and A. Ghrayeb, "A Physics-Aware Framework for Short-Term GPU Power Forecasting of AI Data Centers," in Proc. 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), vol. 2, pp. 4115–4126, 2026.
- [7] N. Enskat and P. Wiesner, "Evaluating MFU as a Proxy for GPU Power for Energy-Aware Simulation of LLM Training," arXiv:2608.03880, 2026; accepted at MASCOTS 2026.
- [8] G. Bolet, G. Georgakoudis, H. Menon, K. Parasyris, N. Hasabnis, H. Estes, K. W. Cameron, and G. Oren, "Can Large Language Models Predict Parallel Code Performance?" in Proc. Workshop on AI for Systems (AI4Sys '25), co-located with HPDC '25, Art. 41, 2025, doi: 10.1145/3731545.3743645.
- [9] M.-K. Nguyen-Nhat, H. D. N. Do, H. T. Le, and T. T. Dao, "LLMPerf: GPU Performance Modeling meets Large Language Models," in Proc. 32nd International Symposium on the Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), pp. 1–8, 2024, doi: 10.1109/MASCOTS64422.2024.10786558.
- [10] T. Tan and G. Cao, "Thermal-Aware Scheduling for Deep Learning on Mobile Devices With NPU," *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 10706-10719, 2024, doi: 10.1109/TMC.2024.3379501.
- [11] M. Smith, L. Zhao, J. Cordova, X. Jiang, and M. Ebrahimi, "Machine Learning-Based Energy-efficient Workload Management for Data Centers," in Proc. IEEE 21st Consumer Communications & Networking Conference (CCNC), pp. 799–802, 2024, doi: 10.1109/CCNC51664.2024.10454842.
- [12] G. Koszczal et al., "Performance and Energy Aware Training of a Deep Neural Network in a Multi-GPU Environment with Power Capping," in Euro-Par 2023: Parallel Processing Workshops, pp. 5-16, 2024, doi: 10.1007/978-3-031-48803-0_1.

- [13] M. Hao et al., "PctoDL: Adaptive GPU Throughput Optimization for Deep Learning Inference with Power Constraints," *ACM Transactions on Architecture and Code Optimization*, vol. 23, no. 2, Art. 43, pp. 1-25, 2026, doi: 10.1145/3805802.
- [14] Y. Wang, Q. Guo, and M. Chen, "Providing load flexibility by reshaping power profiles of large language model workloads," *Advances in Applied Energy*, vol. 19, Art. 100232, 2025, doi: 10.1016/j.adapen.2025.100232.
- [15] J. Aguirre-Usandizaga, A. I. Torre-Bastida, M. Linares de la Puerta, and A. Almeida, "A forecasting-based perception layer for energy-aware resource management in LLM serving deployed on high-performance computing clusters," *Sustainable Computing: Informatics and Systems*, vol. 50, Art. 101333, 2026, doi: 10.1016/j.suscom.2026.101333.
- [16] A. Krzywaniak et al., "Dynamic GPU power capping with online performance tracing for energy efficient GPU computing using DEPO tool," *Future Generation Computer Systems*, vol. 145, pp. 396-414, 2023, doi: 10.1016/j.future.2023.03.041.
- [17] L. He, D. Li, C. Jiang, J. Liu, C. Xu, and L. Zhang, "A predictive offline-online hybrid approach to DNN training energy optimization," *Sustainable Computing: Informatics and Systems*, vol. 51, Art. 101415, 2026, doi: 10.1016/j.suscom.2026.101415.
- [18] A. Afzal, M. M. Li, and M. Panzlaff, "Modeling and Chasing the Energy-Efficiency Sweet Spots in Modern GPUs," *arXiv:2607.00819*, 2026; accepted for PPAM 2026, doi: 10.48550/arXiv.2607.00819.