End-to-End Power Management for a 160 MW AI Datacenter: From Planning to Runtime Optimization

Ehsan Ardestani*, Leonardo Piga, Jovan Stojkovic†, Pavan Balaji, Mustafa Ozdal, Mihaela Dimovska, Mikel Jimenez Fernandez, Hao Shen, Devika Vishwanath, Richa Mishra, Melaku Mihret, Carlos Fernandez, Valentin Andrei, Mauricio Cespedes, Julien Prigent, Tyler Graf, James Monahan, Parth Thakar, Ahmed Sharif, Jesik Min, Luka Tadic, Bin Li, Yang Xia, Shobhit Kanaujia, Kaushik Veeraraghavan, Chunqiang Tang
Meta Platforms †Also with The University of Texas at Austin

***Abstract*—Progress toward artificial general intelligence (AGI) is constrained by compute capacity, which is increasingly limited by power availability. Managing that constraint begins before hardware arrives and continues through deployment and production. Effective power management for AI datacenters spans the entire lifecycle: (1) determining GPU power configurations and cluster size while the hardware is still under development; (2) optimizing power settings after deployment using representative workloads; and (3) managing power dynamically at runtime. Prior research has focused largely on runtime management. We report Meta's end-to-end experience across all three phases for a 160 MW cluster comprising 83K GB200 GPUs. Relative to a counterfactual 1200 W design, planning at 960 W projects 6.6% higher aggregate throughput. Deployment measurements support a further 1–2% gain at 1020 W, while production telemetry indicates that Dimmer's 1120 W operating point provides an additional estimated 2–4%. Compounded, these contributions correspond to an estimated 10–13% improvement. We also report production lessons, including targeted power-limit increases to mitigate training stragglers and per-phase control to prevent unnecessary capping under three-phase current imbalance, and we release production power traces from the cluster.**

I. Introduction

Power availability increasingly constrains the expansion of AI datacenters and, consequently, progress toward artificial general intelligence (AGI) [1]–[6]. For example, Meta and OpenAI have announced plans to build tens of gigawatts (GW) of compute with various partners [1]–[4], [7]–[10].

For perspective, total U.S. power-generation capacity increased from 1,189 GW in 2023 to 1,230 GW in 2024, an addition of 41 GW [5], [6]. These announcements illustrate that planned AI infrastructure is comparable in scale to annual additions to U.S. generation capacity.

Prior work largely studies operational mechanisms such as power capping, energy-cost optimization, and renewable-energy use [11]–[13]. These mechanisms are necessary but incomplete for two reasons. First, datacenters operate under fixed utility power limits. When power is the binding capacity constraint, operators must maximize aggregate compute within that limit rather than optimize electricity cost alone.

Second, power management spans the entire cluster lifecycle: (1) **planning**, when operators use limited vendor data 6–12 months before deployment to select GPU power limits and determine how many GPUs fit within the facility envelope; (2) **deployment validation**, when measured power and representative workloads refine those initial limits; and (3) **active operation**, when runtime control adapts GPU power limits to workload and infrastructure conditions. We report experience across all three phases for a 160 MW cluster with 83K GB200 GPUs.

We make five contributions:

1) **Planning.** We use projected power–performance curves to select a 960 W GPU power limit, rather than the 1200 W TDP, and deploy 83K GPUs. Relative to a counterfactual 1200 W design, this configuration projects 6.6% higher aggregate throughput (Section IV). To our knowledge, this is the first reported hyperscale GPU deployment that uses a sub-TDP power limit to optimize cluster-level Perf/Watt . NVIDIA subsequently introduced MAX-Q and MAX-P workload power profiles for GB300, reflecting similar efficiency and performance operating modes [14].
2) **Power–performance validation.** After deployment, we characterize how GB200 compute throughput and high-bandwidth-memory bandwidth respond to GPU power limits, validating the assumptions used during planning (Section IV-B).
3) **Deployment validation.** We show that power supply unit telemetry systematically overestimates rack power, calibrate it against power-device sensors, and quantify headroom stranded by heterogeneous rack placement. These measurements support increasing the initial GPU power limit (Sections V-A and V-B). We release the underlying oscilloscope, PSU, and DCIM power traces.
4) **Active operation.** We present Dimmer, which dynamically clears device-level power overages. Dimmer caps jobs from lowest to highest priority and, within each priority, from smallest to largest, reducing the impact on large synchronous training runs while enabling a higher fleet GPU power limit (Section VI).
5) **Production lessons.** We characterize two deviations from planning assumptions: three-phase current imbalance that triggers capping below an aggregate device-power limit and strands more power than predicted (Section VII-A); and under-torqued 12 V busbar screws that cause GPUs to overreport power, trigger firmware throttling, and create stragglers that determine the pace of synchronous training jobs (Section VII-B).

*Corresponding author: Ehsan Ardestani, ehsanardestani@meta.com.

Together, these results provide an end-to-end account of power planning and operation for a 160 MW AI cluster.

II. System Background and Definitions

Workloads. We focus on synchronous pre-training as the hardest case for power management: all GPUs peak together, and capping one worker slows the whole job. The planning methodology also applies to inference with a different per-GPU performance curve, and we have used it there.

TDP and power limit. TDP denotes the vendor's hardware design rating, which is 1200 W for GB200. A GPU power limit is the software-configured runtime limit, such as 960 W, which cannot exceed the TDP.

Datacenter. The 160 MW AI cluster spans five datacenter buildings, each provisioned for 32 MW of IT power, within Prometheus, Meta's larger 1 GW AI supercluster [15]. Each building contains four data halls, with three main switchboards (MSBs) supplying each hall.

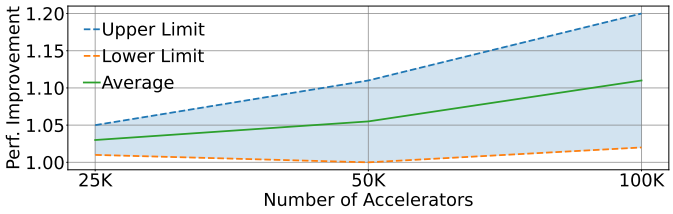


Fig. 1: *(Projected)* Performance improvement of pre-training on GB200 with 100 GB/s per-GPU bandwidth over a 50 GB/s per-GPU bandwidth as cluster size varies.

Rack. NVIDIA's GB200 reference design provides 50 GB/s of scale-out RDMA bandwidth per GPU. This is the same as the Hopper generation, while compute has considerably improved in Blackwell generation. Our simulation [16] showed up to a 20% performance improvement for a 100K GPU cluster with double the scale out bandwidth, as presented in Figure 1. Hence, our Catalina rack design [17] provides 100 GB/s of backend network bandwidth per GPU, twice the 50 GB/s in NVIDIA's GB200 reference design. Because 100 GB/s CX8 was not yet available, each GPU uses two 50 GB/s CX7 NICs. To support that, Grace CPU I/O limitations require one dedicated CPU per GPU, resulting in a 2:2 GPU-to-CPU ratio instead of the 4:2 ratio in the reference design. This leads to a rack containing 36 GB200 GPUs. Trunk cables connect the NVSwitches across two racks, creating a 72-GPU NVLink scale-up domain in this 2x36 configuration that matches the reference design's 1x72 configuration.

Cooling. GB200's 1200 W TDP necessitates liquid cooling to maintain safe and efficient operation. In a Catalina pod, each compute rack is paired with two Air-Assisted Liquid Cooling (AALC) racks that provide rack-local liquid cooling with air-assisted heat rejection.

Performance Metric. We consider 2 performance metrics. 1) Throughput: This is the throughput of cluster when number of GPUs per job $\ll$ cluster size. $Cluster(GPUCount) \times GPU(flops)$ is a good proxy for throughput. 2) Latency: This is the performance of a single pre-training job spanning the whole cluster. Evaluating the trade offs for the latency is complex, and we use detailed simulation [16]. We aim to improve both metrics.

Data provenance. Each figure caption states whether its data is *projected* (simulation or model), *derived* (computed from planned rack budgets), or *measured* (lab or production telemetry).

III. Cluster and Power Infrastructures

A. Power Delivery Hierarchy and Constraints

Power Distribution Architecture. Power flows from an MSB through intermediate switchboards (SBs) to remote power panels (RPPs), which directly feed racks (Figure 2). Every device imposes a local constraint; unused capacity under one branch cannot compensate for an overloaded sibling branch.

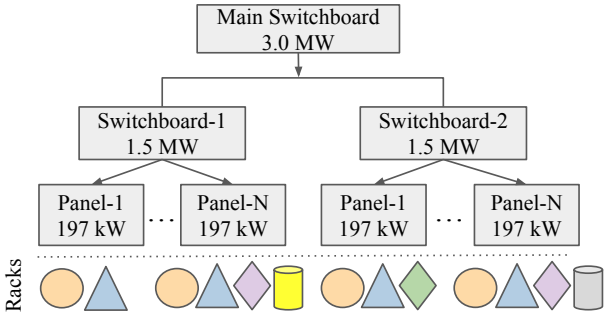


Fig. 2: Power distribution hierarchy. Each shape is a different rack type, highlighting the heterogeneous hardware mix and, thus, the varying power headroom across power domains.

Ratings, Budgets, and Operating Thresholds. Each RPP is protected by a breaker whose continuous per-phase AC current limit is $I_{lim} = 240\,\text{A}$, corresponding to 80% of the breaker rating under the NEC continuous-load rule [18]. At a line-to-line voltage of $V_{LL} = 480\,\text{V}$ and a power factor of 0.99, this current limit corresponds to:

$$P_{lim} = \sqrt{3} V_{LL} I_{lim} \text{PF} \approx 197.5\ \text{kW}. \quad (1)$$

Because each breaker pole protects one AC phase, either the current on a single phase or aggregate power can become the binding constraint.

A typical MSB is rated at 3 MW. Planning allocates 2.7 MW to IT and 300 kW to mechanical systems. This split is flexible as long as their combined load remains within the device envelope. Cooling demand varies with ambient conditions and can exceed the planning allocation during hot periods (Figure 3), so runtime control must observe total MSB load rather than enforce a fixed IT-only budget.

We oversubscribe the hierarchy by allowing downstream nameplate capacity to exceed an upstream device rating. Placement keeps planned rack budgets within upstream allocations, while runtime control keeps measured current and power within operating thresholds. The following sections explain how we choose the initial allocation, validate it after deployment, and manage it in production.

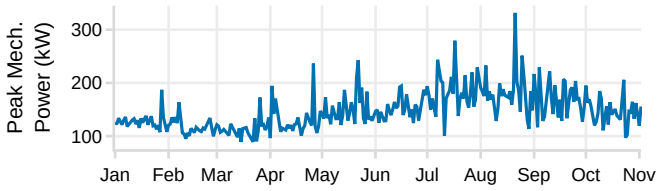


Fig. 3: *(Measured)* Daily mechanical peak one-minute power for an MSB over a 10-month period. Increased cooling demand during warmer summer days occasionally causes power usage to exceed the 300 kW limit estimated during planning.

IV. Phase 1: Planning and Provisioning

Planning must choose a GPU power limit before production hardware and workloads are available. A higher limit improves each GPU's performance but reduces the number of GPUs that fit within the facility budget. We evaluate the two performance objectives defined in Section II. For aggregate throughput, we optimize the product of deployable GPU count and per-GPU performance using the model below. Whole-cluster job latency depends on workload structure, communication, and scaling efficiency, so we evaluate it separately using detailed cluster and workload simulation.

A. Model and Methodology

Let p be the per-GPU power limit, $f(p)$ the projected per-GPU workload throughput, and $g(p)$ the datacenter power attributable to one deployed GPU, including its share of fixed rack and network power. We model

$$g(p) = \delta\left(p + \frac{P_{\text{other}}}{n_r}\right) + P_{\text{net}} + P_{\text{ss}}, \tag{2}$$

where P_{other} is non-GPU rack power, $n_r = 36$ GPUs per rack, P_{net} and P_{ss} are Network and Supporting Services power amortized per GPU. δ is the rack-level derating factor relating configured component power to peak sustainable compute-rack power under representative workloads. The deployable GPU count is

$$N(p) = \min\left(\left\lfloor \frac{P_{\text{total}}}{g(p)} \right\rfloor, N_{\max}\right), \tag{3}$$

where P_{total} is the cluster power budget and $N_{\max}$ captures non-power constraints such as switch radix and floor layout. Our aggregate-throughput objective is

$$\max_{p\in[p_{\min},p_{\max}]} T(p) = N(p)f(p). \tag{4}$$

When power alone determines $N(p)$, this objective is equivalent to maximizing $f(p)/g(p)$. In the deployed design, however, network and layout constraints also bind. We therefore evaluate discrete, feasible rack layouts at every level of the MSB-SB-RPP hierarchy rather than selecting a power limit from Perf/Watt alone. The model separates the accelerator from the facility: another deployment supplies its own $f(p)$ and $g(p)$. It assumes identical GPUs, a fixed portfolio of representative workloads, and a rack-level derating factor δ calibrated to those workloads.

B. Exploring Power Configuration

At planning time, we estimate $f(p)$ with component characterization and an internal graph-execution simulator for representative pre-training workloads [16]. Figure 4 shows empirical data post deployment validating modeling assumptions. The results show that compute throughput becomes power-sensitive primarily at high arithmetic intensity. Figure 5 shows the corresponding HBM-bandwidth sensitivity. Workload phases that do not saturate compute retain steady performance as the power limit falls.

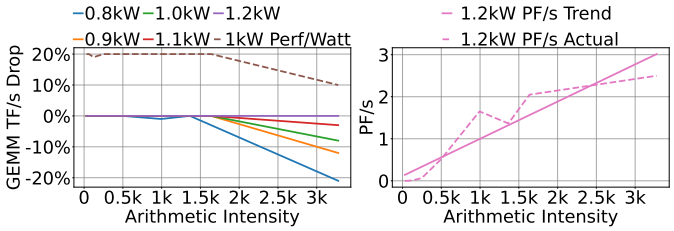


Fig. 4: *(Measured)* GB200 FLOPS sensitivity to power limit. Arithmetic intensity is FLOPS per operand byte.

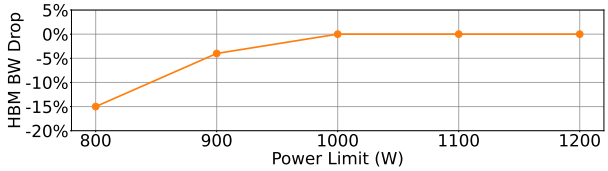


Fig. 5: *(Measured)* GB200 HBM bandwidth sensitivity to power limit.

The simulated end-to-end workload curve is sublinear near 1200 W (Figure 6). Reducing the limit from 1200 to 1000 W lowers per-GPU performance by about 4% while reducing GPU power by 16.7%. At 900 W, per-GPU performance falls by about 12% while GPU power falls by 25%. The additional GPUs enabled by either setting more than offset the per-GPU loss: the power-only model projects 9% higher aggregate throughput at 1000 W and 6% at 900 W relative to 1200 W. Models that stress compute, communication, and memory bandwidth differently have different optima. Because the limit is set 6–12 months before the workloads exist and outlives any model generation, we optimize over a portfolio of pre-training workloads; the throughput curve is broad near its peak. The planning gain is also our most robust result because it comes mostly from GPU count rather than per-GPU performance.

We select 960 W because it lies near the modeled throughput optimum and produces a feasible layout that fully uses backend-network switch radix. This led to 83K GPUs in the cluster, representing 6.6% higher projected aggregate throughput compared to the baseline at 1200 W. Table III summarizes the comparison. Although specific curves depend on the hardware, this methodology is generalizable and has been applied to AMD GPUs and our internal AI accelerator.

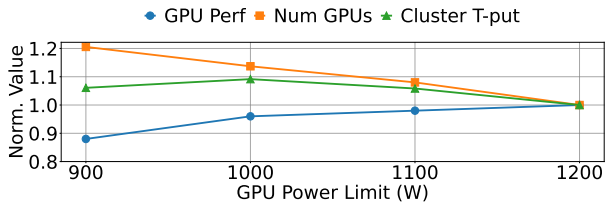


Fig. 6: *(Projected)* At a fixed power budget, higher GPU power increases per-GPU performance but not overall cluster throughput.

Modeling Rack Power Consumption. To estimate rack power consumption, we model the peak sustainable rack power as the sum of the power drawn by each major rack component, with each component's power multiplied by a utilization factor that reflects typical workload behavior. These utilization factors are based on empirical measurements and projections.

We apply a configurable global de-rating factor because rack components rarely peak simultaneously. Representative pre-training simulations, validated against historical telemetry at the lowest power-delivery level, show that GB200 racks draw approximately 90% of their configured maximum after GPU power limits are applied. We therefore set $\delta = 0.90$ for GB200; other configurations may differ. Table I summarizes the component power and de-rating assumptions. GPUs typically account for more than 70% of total rack power. Hence, the GPU power setting has the largest impact on the overall rack power consumption.

TABLE I: Catalina-GB200 provisioned rack power. LiC and AiC denote liquid-cooled and air-cooled components. We are unable to disclose the values of cells marked with an asterisk (*) due to confidentiality requirements.

Component	Unit W	Count	Use %	Total W	Notes
Server-level Components					
GPU+HBM	960	36	100%	34560	LiC
CPU+DIMM	100	36	100%	3600	LiC
Back. NIC	*	36	90%	*	LiC, 2x CX7+optics / GPU
Front. NIC	*	36	80%	*	AiC, 1x 200G NIC / GPU
SSD	15	36	60%	324	AiC, 1xE1.S / GPU
Misc	50	18	50%	450	AiC, BMC, Boot Drive, etc.
NVSwitch	580	18	90%	9396	LiC, 2 dies / tray
OSFPs	*	162	90%	*	LiC
Fabric Mg.	*	9	50%	*	AiC
VR Loss	15%			2436	AiC, Non CPU-GPU
Fans	7%			289	
Rack-level Components					
FE + optics	*	2	100%	*	AiC, 2x Wedge400
Mg. logic	*	1	100%	*	AiC
PSU loss	4%			2239	
Total Power (W)					
Total DC	52820				
Total AC	55966				AC-DC conv. loss
Provisioned Power	**49553**				
Max AC	72124				Without derating
AALC Power	3%			1585	

Backend Network Power Allocation. Table II lists the bottom-up estimate of backend-network power based on the number of switches and their average power consumption.

Finalized Provisioned Rack Power. Table III compares datacenter power allocation and resulting cluster performance.

TABLE II: Provisioned backend-network power

Switch tier	Unit kW	Count	Total kW	Notes
RS	1.88	3	5.6	1st layer switch
FS	1.88	0.5	0.9	2nd layer switch
SS	1.99	2.25	4.5	3rd layer switch
BE per two IT racks			11.1	11% of IT racks; 8–9% of cluster power

TABLE III: Planning comparison for two GB200 power limits. Performance is normalized to 1200W GB200.

	960 W	**1200 W**
Provisioned rack power	49.6 kW	57.3 kW
Power-only GPU count	86,000	74,000
Final deployable GPU count	83,000	74,000
Normalized per-GPU performance	0.95	1.0
Normalized aggregate performance	1.066	1.0

V. Phase 2: Deployment Validation

The 960 W planning limit is deliberately conservative because it relies on projected workloads and component power. After deployment, we replace those projections with calibrated telemetry and representative training runs. Two properties determine how much margin can be recovered.

Temporal averaging. Circuit breakers trip based on overdraw duration as well as magnitude, effectively enforcing limits on moving-average power rather than instantaneous spikes. For example, an RPP may tolerate a 10% overdraw for ∼17 minutes but trips in ∼60 seconds at 40%, whereas an MSB may trip in ∼60 seconds at only 15% overdraw [19], [20]. Operational power tuning must respect these time scales.

Spatial averaging. Peak power across GPUs and racks is not perfectly aligned and shows millisecond divergence, even for synchronous pre-training workloads. This lack of simultaneity creates spatial smoothing that allows per-rack power to be provisioned closer to a lower percentile of the aggregate distribution rather than the absolute peak.

A. Calibrating Fleet-Wide Power Telemetry

Rack-level power control requires accurate telemetry available across the fleet. We use PSU-reported AC power, computed from 100 ms readings with a one-second running average and logged every few seconds. To quantify its accuracy, we compare it with high-frequency oscilloscope measurements at the rack input and Data Center Infrastructure Management (DCIM) sensors at the RPP level. PSU readings consistently exceed the oscilloscope measurements (Figure 7).

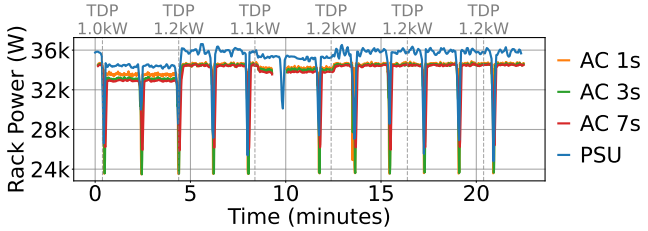


Fig. 7: *(Measured)* PSU versus AC oscilloscope power measurements.

Oscilloscope measurements are accurate but cannot be deployed fleet-wide, while DCIM sensors aggregate several racks. We therefore compare each RPP's DCIM reading with temporal aggregation statistics computed from the PSU readings of its downstream racks (Figure 8).

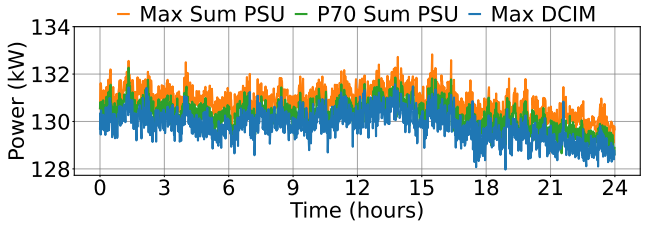


Fig. 8: *(Measured)* Comparison of maximum DCIM power samples with maximum and P70 of the sum of PSU power for one RPP.

The per-minute 70th percentile (P70) of the summed PSU samples minimizes the discrepancy with DCIM, so we use P70 to correct PSU overestimation while retaining fleet-wide coverage.

We open-source the oscilloscope, PSU, and DCIM power traces behind Figures 7, 8, and 10 to support further research on rack-level power telemetry and modeling.[1]

B. Headroom Fragmentation

We define planned power headroom as a device's power limit minus the sum of the rack budgets below it. If identical loads were distributed perfectly, devices at the same hierarchy level would have equal headroom. Physical space, cooling, connectivity, and network-topology constraints instead produce heterogeneous mixes of compute, cooling, network, and service racks, and fragmentation of headroom.

$$PPH = \text{Limit} - \sum_{\text{racks under device}} (\text{provisioned power}_{\text{rack}}) \quad (5)$$

Figure 9 quantifies the resulting fragmentation. More than half of MSBs have at least 160 kW of headroom, but 13% have less than 50 kW. Mean MSB headroom is about 160 kW, or only 100 W per downstream GPU. At the RPP level, mean headroom is 26 kW, more than 200 W per GPU, although a small number of RPPs are nearly full. Thus, MSBs are usually the binding level for a uniform fleet-wide power increase.

Moving service racks can relieve some constrained domains, but moving compute racks after deployment requires disruptive power and network recabling. We observe that up to 10% of cluster power can remain stranded by placement. This fragmentation has two implications: placement must be power-aware during planning, and runtime control must operate per power domain rather than apply one cluster-wide limit.

[1]https://github.com/facebookresearch/meta-datacenter-power-traces

C. Raising the Static GPU Power Limit

We next run a representative Transformer-based model training at small scale of a few racks while sampling rack power every three seconds (Figure 10). Combining this distribution with calibrated rack telemetry shows that a 1020 W GPU limit keeps sustained rack power (moving averages over windows longer than one second) within the 49.6 kW planning budget. We project raising the limit from 960 to 1020 W to improve both the *latency* and *throughput* metrics by 1–2%.

In hindsight, the planning model captured the shape of the trade-off, and its rack power budget proved conservative: $\delta = 0.90$ matched telemetry, and calibrated P70 rack power reached the 49.6 kW budget only at 1020 W. The model missed device variance because it assumes identical GPUs (Section VII-B). The cluster-level curve in Figure 6 cannot be measured directly, because the facility cannot run the fleet above 1120 W without capping.

D. Smoothing Training-Induced Power Swings

Large synchronous jobs create correlated power oscillations that can propagate to the utility interface. The magnitude and frequency of these transients may risk the stability of the power utility grid and may damage the power delivery infrastructure.

We smooth these oscillations by running a low-priority Tensor Core kernel during gaps in the training workload. The kernel operates on register-resident data, avoiding L2 and HBM traffic, and can draw up to 800 W per GB200 when the application is idle.

We considered event-triggered smoothing, but application hooks require intrusive framework changes and power telemetry is too coarse to trigger at the required time scale [21]. We therefore use an always-on kernel with adaptive backoff. The kernel monitors its own instruction latency; when latency indicates contention with training, it yields the affected streaming multiprocessor. This design requires only an environment variable [22], adds less than 3% application overhead in our measurements, and removes the observed pulses (Figure 11). The smoother does not reduce the GPU count: that count is set by provisioned peak rack power, and the smoother fills power valleys without raising the peak, because its 800 W load stays below the GPU power limit and yields to training on contention. Figure 11 shows a test on a production run, prepared in case utilities require smoothing; production jobs on this cluster have not needed it so far.

Hardware alternatives. NVIDIA documents that its hardware power-smoothing circuitry wears with use; `nvidia-smi` reports its remaining lifetime [23]. A Capacitor Backup Unit (CBU) can buffer the same transients but requires dedicated hardware. A software smoother avoids both constraints; whether to rely on it alone or to incur additional hardware cost depends on utility requirements, cluster size, and performance/cost trade-offs.

VI. Phase 3: Active Operation

As established in Section V-B, static headroom fragmentation leaves up to 10% of power stranded across the delivery

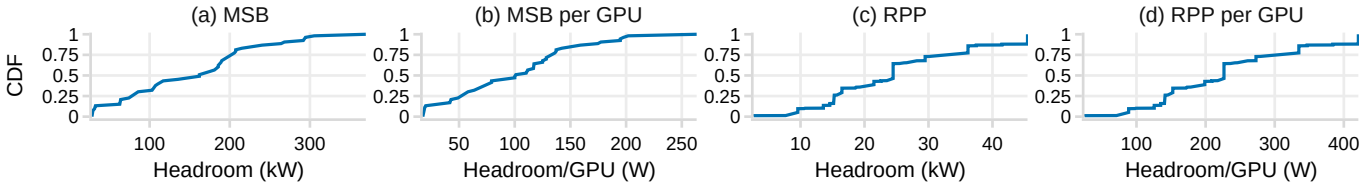


Fig. 9: *(Derived)* Cumulative Distribution Function (CDF) of planned power headroom across MSBs and RPPs: (a) per-MSB, (b) per-GPU at MSB, (c) per-RPP, (d) per-GPU at RPP. Headroom/GPU shows opportunity to increase GPU maximum power limits (in watts) to increase performance.

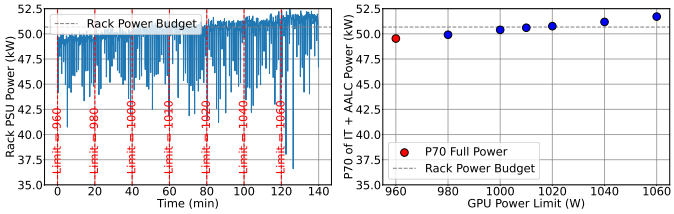


Fig. 10: *(Measured)* Power consumption of a GB200 rack running a dense Transformer model at a given GPU power limit.

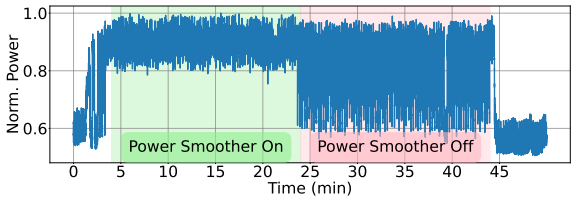


Fig. 11: *(Measured)* A software-based power smoother removing training-related power pulses in a test on a region-scale production training run.

hierarchy. Under a uniform power limit, the most constrained MSB dictates the safe operating range for the *entire* deployment.

For example, in our 160 MW cluster, MSB-A has 50 kW of headroom and is the binding constraint under the static 1020 W policy (Section V-B), while MSB-B retains 310 kW.

More broadly, we estimate that a dynamic heterogeneous power allocation that can recover the static headroom fragmentation can recover more than 3% of cluster throughput. This motivates **Dimmer** [15], an AI workload-aware policy built on top of Dynamo [20]. Dimmer reuses Dynamo's telemetry, power-delivery hierarchy model, controller and monitoring infrastructure, host actuation, and established reliability and operational safeguards, and therefore inherits Dynamo's failure-handling mechanisms for the shared substrate. It adds synchronous-training job ordering, current-aware triggering, and a GPU-limit policy. Priority-ordered capping also exists in prior work [24]–[26]; Dimmer's main adaptation for synchronous training is ordering caps by job size within each priority, because capping one GPU slows its whole job. Its role in this paper is to enable the 1120 W operating point safely. This section explains the Dimmer-specific policy.

A. Design Principles

Dimmer follows four principles derived from synchronous training behavior. First, it degrades performance gradually instead of evicting jobs or abruptly removing large amounts of power [20], [25]. Second, it protects large jobs: capping one worker slows every worker in a synchronous job, so the cost of touching a job grows with job size (Figure 12). Dimmer therefore orders jobs by priority and then by size, capping the smallest low-priority jobs first. Third, it filters transient spikes with a seven-second moving average, which remains well within breaker trip times. Fourth, it caps one host at a time, lowering its GPU power limit to 800 W, until the overage clears, and automatically releases each cap after seven minutes. The low capped limit buys time before a breaker trips in severe events such as the loss of a phase.

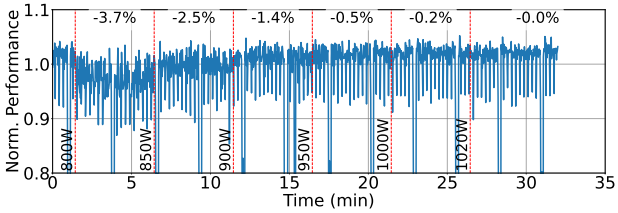


Fig. 12: *(Measured)* Adjusting power limit for a host in a training job, and the indirect impact on job performance and power draw. The y-axis is normalized performance of the job excluding the host for which the power limit changes.

B. System Architecture and Algorithm

Power sensors at RPPs, SBs, and MSBs are polled as frequently as every three seconds through Modbus, BACnet, and OPC-UA [27]–[29]. A centralized controller queries the telemetry service, computes seven-second moving averages, and independently evaluates every power device.

Each three-phase breaker has one pole per phase, so the maximum phase current, not aggregate power, can be the binding constraint. Dimmer maps the maximum measured current I_{meas_max} into an equivalent control power:

$$P_{derived} = \frac{I_{meas_max}}{I_{lim}} P_{lim}. \tag{6}$$

When $I_{meas_max} = I_{lim}$, the derived power reaches the device power limit. This conversion lets one control loop protect both per-phase current and aggregate power. Dimmer uses current-derived power when phase telemetry is available and measured device power otherwise. For RPPs, we set the capping thresholds just below both ceilings: $\alpha_c = 0.99$ (237.6 A per phase) and $\alpha_p = 0.97$ (191.5 kW). Algorithm 1 gives the complete loop. The controller runs independently for each power device (RPP, SB, MSB).

Algorithm 1 Dimmer Algorithm for a Power Device

```
while system is running do
    if per-phase currents available then
        ctrlPwr ← (I_meas_max / I_lim) × P_lim {current-derived power}
        α ← α_c {α_c is the current capping threshold}
    else
        ctrlPwr ← measuredDvcPwr {power fallback}
        α ← α_p {α_p is the power capping threshold}
    end if
    if ctrlPwr > α × P_lim then
        pwrToReclaim ← ctrlPwr − α × P_lim
        for pri in sorted(priorities) do
            for s in group_pri do
                if pwrToReclaim ≤ 0 then
                    break
                end if
                if isCapped(s) then
                    continue
                end if
                serverCapList.append([sId, cappedLimit])
                dec(pwrToReclaim, avgPwr_s − serverMinPwrCapped)
            end for
            if pwrToReclaim ≤ 0 then
                break
            end if
        end for
        capTime ← currentTime
        PwrCapHosts(serverCapList) {set GPU power limit ← cappedTdp}
    else if capTime + capExpirationTime < currentTime then
        capTime ← ∞
        uncapHosts(serverCapList)
    end if
    sleep(Δt)
end while
```

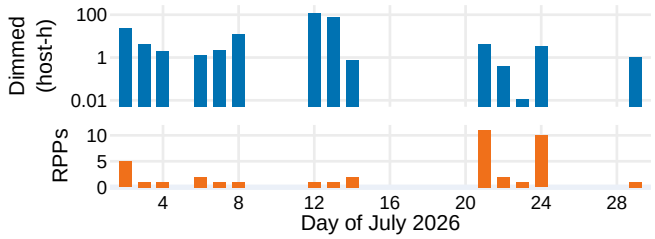


Fig. 13: *(Measured)* Dimmer engagement on the GB200 cluster over July 2026. Top: host-hours held at reduced power limit each day, log scale. Bottom: distinct RPPs engaged.

C. Deployment Results

Dimmer allows GPUs to run at a higher power limit than is possible with static configurations. Figure 13 reports its production activity during July 2026. Dimmer engaged on 14 calendar days, affecting 97 hosts across 20 RPPs and totaling 248.1 capped host-hours. After taking the union of all capping intervals across the cluster, Dimmer was active for less than 0.1% of the month's wall-clock time. We report wall-clock engagement because capping even one GPU can slow its entire synchronous training job; capped GPU-hours would therefore understate the performance exposure. During this period, the fleet base power limit was 1120 W, compared with the 1020 W static baseline (Section V-C). Reaching 1120 W also required raising the per-phase current ceiling to stop imbalance-driven capping (Section VII-A), with Dimmer as the safety net for any remaining overages. Every capping event is logged and shown in job logs and capping dashboards. Capped jobs keep running at reduced, predictable throughput; the scheduler does not yet adjust completion-time estimates for capping, which we leave to future work. Capping affected less than 0.01% of the month's GPU-hours.

TABLE IV: Aggregate-throughput gains across the three phases.

Phase	Baseline and change	Gain	Evidence
Planning	1200 W design → 960 W, 83K GPUs	6.6%	Projected
Deployment	960 W fleet → 1020 W	1–2%	Measured curve
Operation	Uniform 1020 W → Dimmer at 1120 W	2–4%	Telemetry estimate
Compounded end-to-end estimate		**10–13%**	Mixed evidence

An A/B test of Dimmer in the production environment was not feasible. July 2026 telemetry indicates that a uniform static policy would have needed to reserve approximately 8 kW per RPP to keep every RPP within its limit. This is equivalent to approximately 100 W per GPU (9%) under the affected devices. Based on the measured power–performance curves, avoiding this uniform reserve corresponds to an estimated 2–4% aggregate-throughput gain and a 2–3% single-job latency gain relative to the 1020 W static policy. The gain measured on our largest job was higher because the extra power also accelerated throttled stragglers (Section VII-B); we exclude that effect from the estimate.

Putting it all together. Table IV reports the baseline and evidence for each phase. The gains are sequential, so we compound rather than add them. The resulting 10–13% end-to-end range is an estimate that combines projected planning results, deployment measurements, and production telemetry; it is not a single A/B measurement. Telemetry calibration (Section V-A) has no separate row: it adds no throughput by itself, but correcting the PSU overestimate revealed the headroom for 1020 W, so its value appears in the Deployment row.

Which constraint binds. The active constraint shifts as the power limit rises. At 960 W, backend-network switch radix and floor layout cap the cluster at 83K GPUs (86K on power alone; Table III), leaving power that deployment converts into a higher limit. At 1020 W, rack power binds (Figure 10), and the most constrained MSB sets the uniform limit (Section V-B). At 1120 W with Dimmer, the heaviest RPP phase current binds while total RPP power stays below its limit (Figure 14). Beyond that, GPUs that misreport their power throttle themselves (Section VII-B).

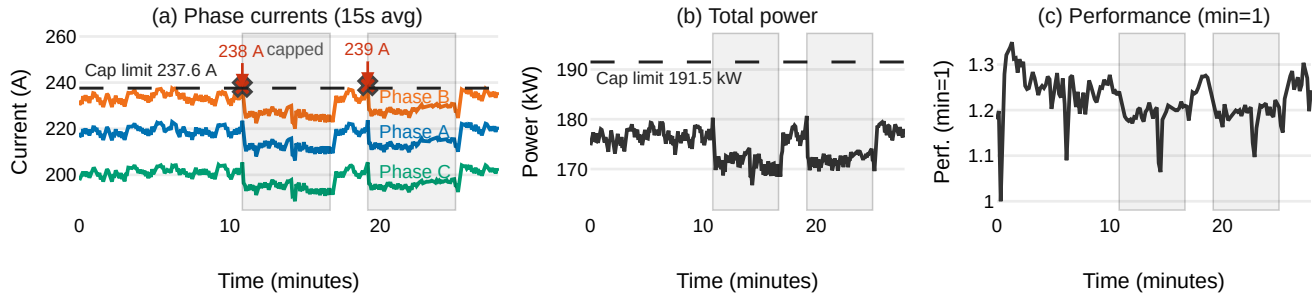


Fig. 14: *(Measured)* Imbalance-driven capping in an RPP. The heaviest phase (B) hits the 237.6 A per-phase limit, resulting in capping followed by dropping throughput in shaded windows, while total RPP power remains under the 191.5 kW limit.

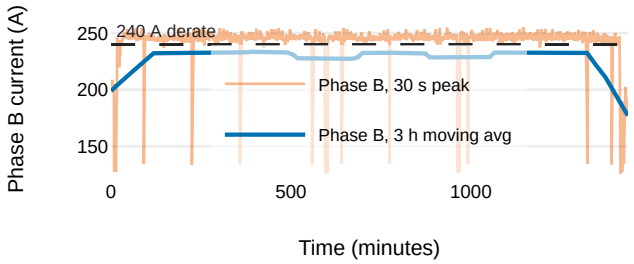


Fig. 15: *(Measured)* Sustained versus instantaneous phase current. The 3-hour moving average of the worst-case phase stays below the 240 A derate, which is what the derate actually governs.

VII. Power-Related Production Issues

The planning phase model in Section IV assumes an ideal cluster, ignoring practical issues like datacenter AC current phase imbalance or device performance variability at a fixed power limit [30]. This section details two production issues that caused stranded power and lost performance, along with our mitigation strategies, and one major learning: while we should plan for a Perf/Watt optimized power limit, we must retain the capability to raise the power limits of individual GPUs above the intended fleet average as an indirect measure against process and manufacturing variation.

A. Three-Phase Current Imbalance Within an RPP

The datacenter distributes three-phase AC power to racks. Provisioning assumes balanced loads across all three phases as a simplification. However, current distributes unevenly in practice. Because the protective breaker has a dedicated pole per phase, the heaviest phase determines the power availability.

Two sources drive phase imbalance. The first is correctable: single-phase support racks are not always assigned evenly across the phases. The second is inherent to the infrastructure: differences in cable length, phase and neutral conductor arrangement, busbar geometry, cable terminations, and enclosure type can create unequal impedance among phase conductors, resulting in uneven phase currents.

Across the RPPs, the heaviest phase runs 4% above the mean current, costing nearly an equivalent fraction of usable power: the RPP reaches its limit delivering only 96% of its 197.5 kW budget, based on the three-phase average.

Section VI-B explains that the Dimmer controller caps against limits set just below both ceilings: 237.6 A per phase (99% of the 240 A derate) and 191.5 kW total (97% of the 197.5 kW budget). Figure 14 captures this on one of our RPPs during a production training run. The heaviest phase rides at the 237.6 A per-phase capping limit. Total RPP power stays below the 191.5 kW power capping limit, confirming that the binding constraint is one phase rather than the RPP as a whole. Each time the heaviest phase reaches the limit, the controller dims the GPUs under the RPP. Aggregate job throughput drops in step with the shaded capping windows.

The figure captures two cap-and-release cycles in about half an hour with no sign of settling. The oscillation is the real cost and makes performance estimations unpredictable. Dimming never reclaims the stranded headroom; it converts that headroom into repeated throughput loss, and the job pays a performance penalty on every cycle.

Reclaiming stranded headroom requires either lowering the uniform GPU power limit to reduce demand on the heaviest phase or raising the per-phase ceiling to increase supply. Since lowering the power limit reduces performance, we raised the ceiling above the 240 A derate. We concluded that this is permissible because the breaker frame is rated at 300 A, and the NEC's 80% continuous load cap allows brief excursions above 240 A if the three-hour moving average of the worst-case phase remains below the derate, which is the case as shown in Figure 15.

Raising the ceiling stopped the imbalance-driven capping. Together with Dimmer as the safety net for remaining overages (Section VI), it let us lift the uniform GPU power limit from 1020 W to 1120 W, yielding a measured 7.9% performance gain on our largest job. This gain far exceeds the 2–3% latency gain projected from the power–performance curve (Section VI-C), because at 1020 W the cards already operate in the nonlinear region of the frequency–power curve. On further inspection, we note that most of the gain came from a slow

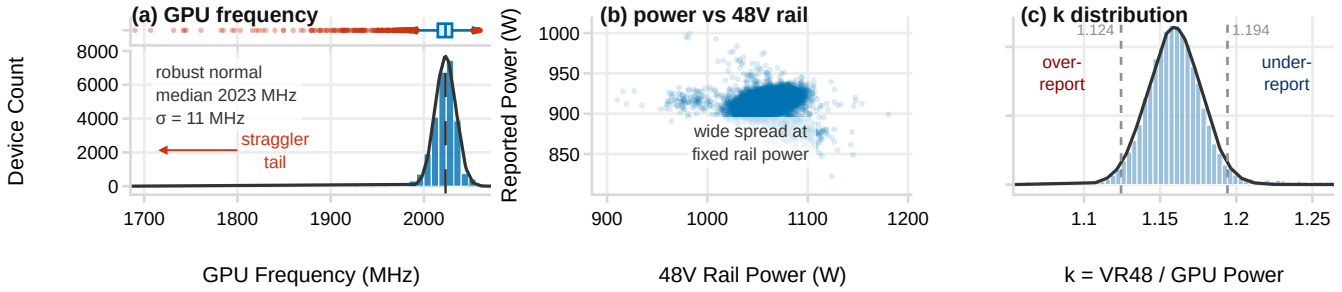


Fig. 16: *(Measured)* Symptom, first clue, and the k metric for GPU device 0 at a fixed 1120 W power limit under one synchronous training job. (a) GPU clock frequency with the robust-normal fit. (b) GPU-reported power against the independently measured 48 V rail power. (c) Distribution of $k = P_{48\mathrm{V}}/P_{\mathrm{GPU}}$ with the fitted robust normal.

tail of hosts that sped up frequency far more than the rest. The next subsection traces this variability to how GPUs can misreport their own power, causing training jobs to slow down.

B. Host Performance Variability at a Fixed Power Limit

Ideally, all GB200 GPUs perform identically at the same power limit. However, on a synchronous training job, we observed a long tail frequency distribution reaching below 1700 MHz compared to ≈2000 MHz expected (Figure 16(a)). This tail, highlighted by outliers in the box plot, dictates the entire run's pace since synchronous training is bound by the slowest host.

The variability pointed to hardware rather than the workload. For the same job we compared each GPU's self-reported power against its 48 V rail power, an upstream feed measured independently of the GPU. Figure 16(b) plots the two. At a given rail power the reported power spreads widely, wider than identical cards running one workload should. Some GPUs claim far more power than their rail delivers.

Root cause: under-torqued busbar screws. We traced performance tails to under-torqued 12 V busbar screws. The screws are not the conductor; their torque sets how firmly the busbar is clamped to the board. With too little clamping force, the joint impedance falls out of specification, part of the current bypasses the power-sensing resistor, and the reported power diverges from the actual draw. The firmware then throttles the GPU below its configured power limit, believing it is over budget even though the actual draw is within limits and the card turns into a straggler. Performance variation at this scale need not originate in silicon: a screw torqued slightly out of spec is enough to slow an entire multi-thousand-GPU run.

Using the rail ratio $k = P_{48\mathrm{V}}/P_{\mathrm{GPU}}$, healthy cards cluster near $k \approx 1.16$, while over-reporters show lower values. Robust normal distributions (median and Median Absolute Deviation) fit to k and GPU frequency classify hosts via $\pm 2\sigma$ thresholds (Figures 16(c), 17). The resulting nine-class grid identifies healthy cards, throttled over-reporters, and under-reporting power risks. Over an 11-hour window across 26 K hosts, the classifier categorizes 76% as healthy, 9% as candidates for power-limit adjustments, and 1.6% (417 hosts) as throttled over-reporting stragglers.

Mitigation: bump the straggler power limit. A hardware fix is slow. As a software bridge we raise the GPU power limit of the identified stragglers from 1120 W to 1200 W, about

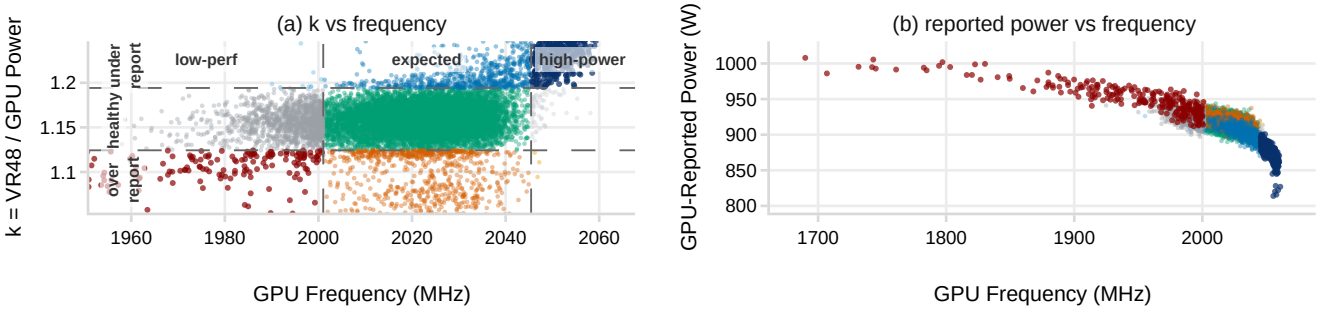


Fig. 17: *(Measured)* Straggler classifier for GPU device 0 on the 48 V rail. (a) Nine-class grid tracking report health (k) against GPU frequency, with $\pm 2\sigma$ cuts (dashed). The over-report/low-performance corner indicates a throttled straggler. Colors represent report health: red (over-report), blue (under-report), gray (normal), and green (healthy center), highlighting actionable extremes. (b) GPU power vs. frequency for the same classes; stragglers show low frequency at high reported power.

80 W of extra headroom that lets an over-reporting card's real draw climb back toward its intended power. On the production job, promoting only the worst stragglers (bottom-left quadrant) lifted median performance by 1.7% over the baseline where every host sits at 1120 W. Across our tests, nearly every promoted card gained frequency from the 1120 W $\rightarrow$ 1200 W bump, a median of about 40 MHz, and the gain grows as the starting frequency falls: the slowest cards (the ones the firmware throttled hardest) recover the most, while cards already near the 2062 MHz clock ceiling have no room left and gain little.

The synchronous training performance is still limited by its slowest host. A larger opportunity would be to remove the defective over-reporting hosts from production entirely rather than compensate them with extra power. However, even after fixing hardware for this particular issue, the performance variation will be present, although not at such degree. We expect dynamic power management to compensate for such variations by allowing for straggler GPUs drawing lower power to go beyond the provisioned power limit. Increasing the power limit for the stragglers does not increase the overall cluster power beyond what was provisioned. It, in fact, prevents otherwise under utilization of provisioned power.

VIII. Takeaways: Insights and Lessons

Drawing on our experience managing the 160 MW, 83K-GPU cluster, this section summarizes key lessons. The methodology transfers, while the operating points (960/1020/1120 W, 4% phase imbalance, up to 10% stranded power) are specific to this system. The production issues also recur elsewhere: phase imbalance in our other deployments, PSU over-reporting in other rack types, and busbar misreporting on GB300 at a much smaller scale.

A. Use a power limit lower than TDP to maximize Perf/Watt at the cluster level

Trading per-device performance for device count under a fixed power budget is well established, and sub-TDP provisioning is common in CPU fleets [20], [31]. Our contribution is applying it at the scale of a GPU fleet: to our knowledge, no hyperscaler or cloud provider had publicly reported provisioning GPUs on a cluster-level Perf/Watt criterion. Under a fixed cluster power budget, we plan GB200 GPUs at 960 W instead of the default setting of 1200 W, which allows us to utilize more GPUs to improve the cluster's aggregate throughput. In contrast, we have not found other hyperscalers or cloud service providers operating GB200s below the NVIDIA-specified GPU TDP of 1200 W, after evaluating public information from Amazon Web Services (AWS) [32], [33], Microsoft Azure [21], Google Cloud Platform (GCP) [34], [35], Oracle Cloud Infrastructure (OCI) [36], CoreWeave [37]–[39], Hyperstack [40], and Spheron [41], [42]. At the time of our GB200 deployment, provisioning GPUs below TDP for Perf/Watt was therefore not common practice. It has since become mainstream: beginning with GB300, NVIDIA offers workload power profiles in which MAX-Q represents the Perf/Watt-optimized operating point and MAX-P the maximum-performance point [14].

At the current stage of the AGI race, power is the most constraining resource for most hyperscalers for the foreseeable future. Therefore, our approach of maximizing cluster-level Perf/Watt may be applicable to others as well.

Although this paper focuses on applying our approach to a GB200 training cluster, we have also utilized it for both training and inference on AMD GPU clusters and those using our proprietary AI accelerators (MTIA [43]).

B. Power management can serve as an effective tool for mitigating stragglers in synchronous training jobs

In synchronous training, a single underperforming GPU can slow down the whole cluster. While GPUs often exhibit performance variations due to manufacturing issues at the silicon or system level, the source of our acute variation was traced to the rack busbar. By increasing the power limit for the impacted GPUs, we enabled them to match the performance of their counterparts, thereby accelerating the overall training job.

More broadly, we believe that stragglers stemming from silicon-level manufacturing variations can be similarly addressed by adjusting power limits. Therefore, designing the cluster for a power limit lower than maximum does not mean locking the power limit per GPU. This represents an ongoing direction of our current research.

C. Three-phase AC current imbalance causes premature breaker capping on individual phase current rather than aggregate power

Our study reveals that current imbalances in three-phase AC delivery arise due to electrical geometry (e.g., cable length and conductor arrangement), where the most heavily loaded phase limits usable power. Therefore, effective power management must be per-phase current-aware, rather than based solely on aggregate power across three phases. We believe this finding applies broadly to high-density AI datacenters. While prior studies have consistently overlooked this issue, we address it by leveraging the fact that the breaker is rated at a higher limit and power delivery allows brief current excursions, as long as the moving average remains below the derate.

D. The accuracy of hardware-supplied power telemetry data should not be taken for granted

Accurate power telemetry is essential for effective power management, yet our experience shows that hardware-reported telemetry can be unreliable for several reasons. Power and rack devices from various vendors may calibrate their telemetry data for use cases different from ours. For instance, the power supply units (PSUs) in our GB200 racks consistently overreport power draw, likely due to the vendor adding a safety margin. In another case, manufacturing defects in rack busbars have led to a subset of GPUs overreporting their power consumption. Therefore, while most research focuses on software solutions in power management, we emphasize

that operators must proactively both calibrate for systematic telemetry bias and detect outliers resulting from manufacturing variations or defects.

E. Prioritize deployability over theoretical elegance: Always-on power smoothing with adaptive backoff

Synchronized training at a datacenter scale generates massive power oscillations that can jeopardize grid stability and power delivery infrastructure. While two theoretically cleaner alternatives exist, they prove difficult to deploy: (1) software-level, event-triggered smoothing [21] requires intrusive modifications to the training software or the PyTorch stack; and (2) power-telemetry-triggered smoothing is often too coarse-grained and reactive to respond effectively. In contrast, our "always-on" approach—utilizing a register-only Tensor Core synthetic load (800W/GPU) with latency-based adaptive backoff—is effective while requiring no application-level changes. A key lesson is that large-scale production systems must prioritize real-world deployability over theoretical elegance.

F. Rack placement is a form of power management; average power headroom is misleading and unusable

To the best of our knowledge, our work is the first to quantify the power headroom imbalance created by the heterogeneous placement of GPU racks, AALC racks, and network devices, as well as the storage racks and CPU-server racks needed to support the GPU racks. Specifically, in our cluster, this results in 13% of MSBs having less than 50 kW headroom, despite a mean headroom of 160 kW (100 W/GPU). While prior studies often assume uniform power headroom when performing runtime power management, in reality, this imbalance strands certain headroom power during synchronous training, which no runtime controller can reclaim without rack relocation. Therefore, power-aware rack placement must be co-designed with network and cooling in the planning phase.

IX. Related Work

Datacenter Power Management. Prior work has explored datacenter-wide power management strategies to improve utilization and efficiency. For example, Piga et al. [31] propose safe DVFS boosting to increase capacity, while Flex [44] and SmartOClock [45] leverage workload-aware oversubscription to maximize power budgets. SmoothOperator [46] reduces peak power draw by coordinating the timing of synchronous services. Many systems employ statistical multiplexing across servers [20], [24]–[26], [46]–[50] to safely oversubscribe power and reduce costs, with such techniques now deployed in production environments [51]–[53]. These approaches typically require tight cluster-level coordination and deep integration with the software stack.

In contrast, our work addresses both the provisioning and operational phases of power management in large-scale AI training clusters. We focus on practical strategies for ensuring safety, maximizing performance, and adapting to dynamic workload and infrastructure conditions at scale.

Energy-efficiency in Datacenters. Prior work has focused on improving energy efficiency for CPU-centric datacenter workloads [54]–[57], with more recent research examining the energy characteristics of GPU-based applications [58]–[61]. Recent schemes manage energy and power consumption for DNN inference and training [62]–[64] through techniques such as frequency scaling [65]–[71], autoscaling [72], and resource partitioning and mapping [69], [73]. These approaches often target fine-grained control at the device or job level, optimizing for throughput or energy proportionality in DNN workloads.

In contrast, our work addresses the challenges of power management at the scale of entire AI training clusters.

X. Conclusion

Power management for a large AI cluster is a lifecycle problem. For Meta's 160 MW, 83K GB200 cluster, planning at 960 W instead of the 1200 W TDP is projected to yield 6.6% higher aggregate throughput. Deployment measurements supported a 1020 W static limit and an additional 1–2% gain. Dimmer then supported a 1120 W fleet limit through dynamic management; during one month it was active for less than 0.1% of wall-clock time, with an estimated 2–4% benefit over the 1020 W static policy. Compounded, these correspond to an estimated 10–13% throughput improvement over the 1200 W baseline. A few lessons extend beyond this cluster. First, cluster-level throughput, not per-device performance, should determine the planning power limit. Second, average headroom is not usable headroom: physical placement strands up to 10% of power unless planning and runtime control both respect topology. Third, aggregate watts are insufficient for three-phase infrastructure, and device telemetry must be validated against independent measurements.

References

[1] OpenAI, "OpenAI, Oracle, and SoftBank expand Stargate with five new AI data center sites," 2025, https://openai.com/index/five-new-stargate-sites/.

[2] NVIDIA, "OpenAI and NVIDIA Announce Strategic Partnership to Deploy 10 Gigawatts of NVIDIA Systems," 2025, https://nvidianews.nvidia.com/news/openai-and-nvidia-announce-strategic-partnership-to-deploy-10gw-of-nvidia-systems.

[3] AMD, "AMD and OpenAI Announce Strategic Partnership to Deploy 6 Gigawatts of AMD GPUs," 2025, https://ir.amd.com/news-events/press-releases/detail/1260/amd-and-openai-announce-strategic-partnership-to-deploy-6-gigawatts-of-amd-gpus.

[4] OpenAI, "OpenAI and Broadcom announce strategic collaboration to deploy 10 gigawatts of OpenAI-designed AI accelerators," 2025, https://openai.com/index/openai-and-broadcom-announce-strategic-collaboration/.

[5] EIA, "Electricity explained: Electricity generation, capacity, and sales in the United States," 2024, https://www.eia.gov/energyexplained/electricity/electricity-in-the-us-generation-capacity-and-sales.php.

[6] ——, "U.S. battery capacity increased 66% in 2024," 2025, https://www.eia.gov/todayinenergy/detail.php?id=64705.

[7] M. Zuckerberg, "2025 will be a defining year for AI — Meta infrastructure investment announcement," Facebook post, Jan. 2025, announced $60–80B CapEx for 2025, 1.3M GPUs by year-end, and a new data center large enough to cover a significant part of Manhattan. Accessed: 2026-08-01. [Online]. Available: https://www.facebook.com/zuck/posts/10115772700696221

[8] ——, "Meta is building Hyperion — a 5GW AI data center," Threads post, Jul. 2025, announced Hyperion, Meta's multi-gigawatt AI data center cluster scaling to 5GW. Accessed: 2026-08-01. [Online]. Available: https://www.threads.net/@zuck/post/C9dHyXkJ8Wz

[9] ——, "Meta compute — building gigawatt-scale AI infrastructure," Meta announcement, Jan. 2026, announced the Meta Compute initiative to build multi-gigawatt-scale AI computing capacity, including plans for tens of gigawatts this decade. Accessed: 2026-08-01. [Online]. Available: https://about.fb.com/news/2026/01/meta-compute/

[10] "Meta and amd partner for longterm ai infrastructure agreement." [Online]. Available: https://about.fb.com/news/2026/02/meta-amd-partner-longterm-ai-infrastructure-agreement/

[11] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Zomaya, "A taxonomy and survey of energy-efficient data centers and cloud computing systems," *Advances in computers*, vol. 82, pp. 47–111, 2011. [Online]. Available: https://doi.org/10.1016/B978-0-12-385512-1.00003-7

[12] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Communications surveys & tutorials*, vol. 18, no. 1, pp. 732–794, 2015. [Online]. Available: https://doi.org/10.1109/COMST.2015.2481183

[13] L. Benini, A. Bogliolo, and G. De Micheli, "A survey of design techniques for system-level dynamic power management," *IEEE transactions on very large scale integration (VLSI) systems*, vol. 8, no. 3, pp. 299–316, 2002. [Online]. Available: https://doi.org/10.1109/92.845896

[14] NVIDIA, "Optimize data center efficiency for AI and HPC workloads with power profiles," 2025, NVIDIA Technical Blog. [Online]. Available: https://developer.nvidia.com/blog/optimize-data-center-efficiency-for-ai-and-hpc-workloads-with-power-profiles/

[15] K. Lee, Y. Bayani, A. Sharif, and Y. J. Song, "Inside Prometheus: Meta's journey to a gigawatt-scale supercluster," Engineering at Meta blog, Oct. 2026, accessed: 2026-11-03. [Online]. Available: https://engineering.fb.com/2026/10/19/data-center-engineering/inside-prometheus-meta-gigawatt-scale-supercluster/

[16] P. Park, H. Lim, and Seah Kim et al., "AIPerf: Meta's Performance Modeling and HW/SW Co-design Framework for Datacenter-Scale LLM Training," in *59th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2026.

[17] "The open compute project accelerating deployment of next gen ai clusters." [Online]. Available: https://www.opencompute.org/blog/the-open-compute-project-accelerating-deployment-of-next-gen-ai-clusters

[18] National Fire Protection Association, *NFPA 70: National Electrical Code (NEC)*, 2026th ed. Quincy, Massachusetts: National Fire Protection Association, 2025.

[19] X. Fu, J. Guo, Y. Wang, Y. Zhao, M. Lin, F. Liu, Z. Wu, and J. Li, "How much power oversubscription is safe and allowed in data centers?" in *Proceedings of the 8th International Conference on Autonomic Computing (ICAC 2011)*. ACM, 2011, pp. 91–100. [Online]. Available: https://doi.org/10.1145/1998582.1998596

[20] Q. Wu, Q. Deng, L. Ganesh, C.-H. Hsu, Y. Jin, S. Kumar, B. Li, J. Meza, and Y. J. Song, "Dynamo: Facebook's Data Center-Wide Power Management System," in *Proceedings of the 43rd Annual International Symposium on Computer Architecture (ISCA '16)*, 2016. [Online]. Available: https://doi.org/10.1145/3007787.3001187

[21] E. Choukse, B. Warrier, S. Heath, L. Belmont, A. Zhao, H. A. Khan, B. Harry, M. Kappel, R. J. Hewett, K. Datta *et al.*, "Power stabilization for ai training datacenters," *arXiv preprint arXiv:2508.14318*, 2025.

[22] "Power smoother pytorch pull," https://github.com/pytorch/pytorch/pull/132936.

[23] NVIDIA, "NVIDIA System Management Interface (nvidia-smi)," https://docs.nvidia.com/deploy/nvidia-smi/index.html, 2026, accessed July 30, 2026.

[24] V. Sakalkar, V. Kontorinis, D. Landhuis, S. Li, D. De Ronde, T. Blooming, A. Ramesh, J. Kennedy, C. Malone, J. Clidaras, and P. Ranganathan, "Data Center Power Oversubscription with a Medium Voltage Power Plane and Priority-Aware Capping," in *Proceedings of the 25th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '20)*, 2020. [Online]. Available: https://doi.org/10.1145/3373376.3378533

[25] S. Li, X. Wang, X. Zhang, V. Kontorinis, S. Kodakara, D. Lo, and P. Ranganathan, "Thunderbolt: Throughput-Optimized, Quality-of-Service-Aware Power Capping at Scale," in *Proceedings of the 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI '20)*, 2020.

[26] A. G. Kumbhare, R. Azimi, I. Manousakis, A. Bonde, F. Frujeri, N. Mahalingam, P. A. Misra, S. A. Javadi, B. Schroeder, M. Fontoura, and R. Bianchini, "Prediction-Based Power Oversubscription in Cloud Platforms," in *Proceedings of the USENIX Annual Technical Conference (USENIX ATC '21)*, 2021.

[27] G. Thomas, "Introduction to the modbus protocol," *The Extension*, vol. 9, no. 4, pp. 1–4, 2008.

[28] L. K. Haakenstad, "The open protocol standard for computerized building systems: Bacnet," in *Proceedings of the 1999 IEEE International Conference on Control Applications (Cat. No. 99CH36328)*, vol. 2. IEEE, 1999, pp. 1585–1590. [Online]. Available: https://doi.org/10.1109/CCA.1999.801208

[29] O. Foundation, "Unified architecture - opc foundation," https://opcfoundation.org/about/opc-technologies/opc-ua/, Sep. 2019, accessed: Dec. 11, 2025.

[30] M. Kurzynski, S. Aga, and D. Wu, "Lit silicon: A case where thermal imbalance couples concurrent execution in multiple gpus," in *Proceedings of the ACM/IEEE 53rd International Symposium on Computer Architecture (ISCA '26)*, 2026.

[31] L. Piga, I. Narayanan, A. Sundarrajan, M. Skach, Q. Deng, B. Maity, M. Chakkaravarthy, A. Huang, A. Dhanotia, and P. Malani, "Expanding datacenter capacity with dvfs boosting: A safe and scalable deployment experience," in *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*, ser. ASPLOS '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 150–165. [Online]. Available: https://doi.org/10.1145/3617232.3624853

[32] Amazon Web Services, "AWS AI infrastructure with NVIDIA Blackwell: Two powerful compute solutions for the next frontier of AI," 2025, aWS Machine Learning Blog. [Online]. Available: https://aws.amazon.com/blogs/machine-learning/aws-ai-infrastructure-with-nvidia-blackwell-two-powerful-compute-solutions-for-the-next-frontier-of-ai/

[33] ——, "New Amazon EC2 P6e-GB200 UltraServers accelerated by NVIDIA Grace Blackwell GPUs for the highest AI performance," 2025, aWS News Blog. [Online]. Available: https://aws.amazon.com/blogs/aws/new-amazon-ec2-p6e-gb200-ultraservers-powered-by-nvidia-grace-blackwell-gpus-for-the-highest-ai-performance/

[34] Google Cloud, "New A4X VMs powered by NVIDIA GB200 GPUs," 2025, google Cloud Blog. [Online]. Available: https://cloud.google.com/blog/products/compute/new-a4x-vms-powered-by-nvidia-gb200-gpus

[35] ——, "Enabling 1 MW IT racks and liquid cooling at OCP EMEA summit," 2025, google Cloud Blog. [Online]. Available: https://cloud.google.com/blog/topics/systems/enabling-1-mw-it-racks-and-liquid-cooling-at-ocp-emea-summit

[36] Oracle Cloud Infrastructure, "Behind the scenes: Scale your NVIDIA GB200 NVL72 deployments with dedicated OCI APIs," 2025, oracle Blogs. [Online]. Available: https://blogs.oracle.com/cloud-infrastructure/behind-the-scenes-nvidia-gb200-nvl72-oci-apis

[37] CoreWeave, "CoreWeave becomes first NVIDIA GB200 exemplar cloud," 2025. [Online]. Available: https://www.coreweave.com/blog/coreweave-sets-new-standard-as-first-nvidia-gb200-exemplar-cloud-improving-upon-nvidias-own-training-performance-targets

[38] ——, "About GB200 and GB300 NVL72-powered instances," 2025, coreWeave Documentation. [Online]. Available: https://docs.coreweave.com/platform/instances/nvl72

[39] ——, "Why liquid cooling matters for AI scale," 2025. [Online]. Available: https://www.coreweave.com/blog/why-you-need-liquid-cooling-for-ai-performance-at-scale

[40] Hyperstack, "Rent NVIDIA GB200 NVL72 Blackwell GPU: On-demand pricing & specs," 2025. [Online]. Available: https://www.hyperstack.cloud/nvidia-blackwell-gb200

[41] Spheron Network, "GB200 NVL72 guide: Rack specs, cloud price," 2024. [Online]. Available: https://www.spheron.network/blog/nvidia-gb200-nvl72-guide/

[42] NVIDIA, "Chill factor: NVIDIA Blackwell platform boosts water efficiency by over 300x," 2024, nVIDIA Blog. [Online]. Available: https://blogs.nvidia.com/blog/blackwell-platform-water-efficiency-liquid-cooling-data-centers-ai-factories/

[43] MTIA Team, "MTIA-300: Meta's Training Chip with Embedded NIC Chiplets and Communication Offloading Engine," in *International Symposium on Computer Architecture (ISCA)*, 2026.

[44] C. Zhang, A. G. Kumbhare, I. Manousakis, D. Zhang, P. A. Misra, R. Assis, K. Woolcock, N. Mahalingam, B. Warrier, D. Gauthier, L. Kunnath, S. Solomon, O. Morales, M. Fontoura, and

R. Bianchini, "Flex: High-Availability Datacenters with Zero Reserved Power," in *Proceedings of the 48th Annual International Symposium on Computer Architecture (ISCA '21)*, 2021. [Online]. Available: https://doi.org/10.1109/ISCA52012.2021.00033

[45] J. Stojkovic, P. Misra, I. Goiri, S. Whitlock, E. Choukse, M. Das, C. Bansal, J. Lee, Z. Sun, H. Qiu, R. Zimmermann, S. Samal, B. Warrier, A. Raniwala, and R. Bianchini, "SmartOClock: Workload- and Risk-Aware Overclocking in the Cloud," in *Proceedings of the 51st Annual International Symposium on Computer Architecture (ISCA '24)*, 2024. [Online]. Available: https://doi.org/10.1109/ISCA59077.2024.00040

[46] C.-H. Hsu, Q. Deng, J. Mars, and L. Tang, "SmoothOperator: Reducing Power Fragmentation and Improving Power Utilization in Large-Scale Datacenters," in *Proceedings of the 23rd International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '18)*, 2018. [Online]. Available: https://doi.org/10.1145/3296957.3173190

[47] P. Ranganathan, P. Leech, D. Irwin, and J. Chase, "Ensemble-level Power Management for Dense Blade Servers," in *Proceedings of the 33rd Annual International Symposium on Computer Architecture (ISCA '06)*, 2006. [Online]. Available: https://doi.org/10.1109/ISCA.2006.20

[48] S. Govindan, J. Choi, B. Urgaonkar, A. Sivasubramaniam, and A. Baldini, "Statistical profiling-based techniques for effective power provisioning in data centers," in *Proceedings of the 4th ACM European Conference on Computer Systems (EuroSys '09)*, 2009. [Online]. Available: https://doi.org/10.1145/1519065.1519099

[49] A. A. Bhattacharya, D. Culler, A. Kansal, S. Govindan, and S. Sankar, "The need for speed and stability in data center power capping," in *Proceedings of the International Green Computing Conference (IGCC '12)*, 2012. [Online]. Available: https://doi.org/10.1109/IGCC.2012.6322253

[50] V. Kontorinis, L. E. Zhang, B. Aksanli, J. Sampson, H. Homayoun, E. Pettis, D. M. Tullsen, and T. S. Rosing, "Managing Distributed Ups Energy for Effective Power Capping in Data Centers," in *Proceedings of the 39th Annual International Symposium on Computer Architecture (ISCA '12)*, 2012. [Online]. Available: https://doi.org/10.1145/2366231.2337216

[51] Amazon Web Services, "Processor state control for your EC2 instance," April 2024. [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/processor_state_control.html

[52] Microsoft Azure, "Virtual Machine series," April 2024. [Online]. Available: https://azure.microsoft.com/en-us/pricing/details/virtual-machines/series

[53] Google Compute Platform, "CPU platforms," April 2024. [Online]. Available: https://cloud.google.com/compute/docs/cpu-platforms

[54] J. Stojkovic, N. Iliakopoulou, T. Xu, H. Franke, and J. Torrellas, "EcoFaaS: Rethinking the Design of Serverless Environments for Energy Efficiency," in *Proceedings of the 51st Annual International Symposium on Computer Architecture (ISCA '24)*, 2024. [Online]. Available: https://doi.org/10.1109/ISCA59077.2024.00042

[55] R. Nishtala, V. Petrucci, P. Carpenter, and M. Sjalander, "Twig: Multi-Agent Task Management for Colocated Latency-Critical Cloud Services," in *Proceedings of the IEEE International Symposium on High Performance Computer Architecture (HPCA '20)*, 2020. [Online]. Available: https://doi.org/10.1109/HPCA47549.2020.00023

[56] M. E. Haque, Y. He, S. Elnikety, T. D. Nguyen, R. Bianchini, and K. S. McKinley, "Exploiting Heterogeneity for Tail Latency and Energy Efficiency," in *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO '17)*, 2017. [Online]. Available: https://doi.org/10.1145/3123939.3123956

[57] D. Lo, L. Cheng, R. Govindaraju, L. A. Barroso, and C. Kozyrakis, "Towards energy proportionality for large-scale latency-critical workloads," in *Proceedings of the ACM/IEEE 41st International Symposium on Computer Architecture (ISCA '14)*, 2014. [Online]. Available: https://doi.org/10.1109/ISCA.2014.6853237

[58] Z. Tang, Y. Wang, Q. Wang, and X. Chu, "The Impact of GPU DVFS on the Energy and Performance of Deep Learning: An Empirical Study," in *Proceedings of the Tenth ACM International Conference on Future Energy Systems (e-Energy '19)*, 2019. [Online]. Available: https://doi.org/10.1145/3307772.3328315

[59] J. Yu, J. Kim, and E. Seo, "Know Your Enemy To Save Cloud Energy: Energy-Performance Characterization of Machine Learning Serving," in *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2023. [Online]. Available: https://doi.org/10.1109/HPCA56546.2023.10070943

[60] S. Samsi, D. Zhao, J. McDonald, B. Li, A. Michaleas, M. Jones, W. Bergeron, J. Kepner, D. Tiwari, and V. Gadepally, "From words to watts: Benchmarking the energy costs of large language model inference," in *2023 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE, 2023, pp. 1–9. [Online]. Available: https://doi.org/10.1109/HPEC58863.2023.10363447

[61] J. Stojkovic, C. Zhang, Í. Goiri, J. Torrellas, and E. Choukse, "DynamoLLM: Designing LLM Inference Clusters for Performance and Energy Efficiency," in *Proceedings of the IEEE International Symposium on High Performance Computer Architecture (HPCA'25)*, 2025. [Online]. Available: https://doi.org/10.1109/HPCA61900.2025.00102

[62] J. You, J.-W. Chung, and M. Chowdhury, "Zeus: Understanding and optimizing GPU energy consumption of DNN training," in *Proceedings of the 20th USENIX Symposium on Networked Systems Design and Implementation (NSDI '23)*, 2023.

[63] C. Wan, M. Santriaji, E. Rogers, H. Hoffmann, M. Maire, and S. Lu, "ALERT: Accurate learning for energy and timeliness," in *Proceedings of the USENIX Annual Technical Conference (USENIX ATC '20)*, 2020.

[64] F. Wang, W. Zhang, S. Lai, M. Hao, and Z. Wang, "Dynamic GPU Energy Optimization for Machine Learning Training Workloads," *IEEE Transactions on Parallel and Distributed Systems*, 2022. [Online]. Available: https://doi.org/10.1109/TPDS.2021.3137867

[65] S. M. Nabavinejad, S. Reda, and M. Ebrahimi, "Coordinated Batching and DVFS for DNN Inference on GPU Accelerators," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 10, 2022. [Online]. Available: https://doi.org/10.1109/TPDS.2022.3144614

[66] Y. Zhang, Q. Wang, Z. Lin, P. Xu, and B. Wang, "Improving GPU Energy Efficiency through an Application-transparent Frequency Scaling Policy with Performance Assurance," in *Proceedings of the Nineteenth European Conference on Computer Systems (EuroSys '24)*, 2024. [Online]. Available: https://doi.org/10.1145/3627703.3629584

[67] S. M. Nabavinejad, S. Reda, and M. Ebrahimi, "BatchSizer: Power-Performance Trade-off for DNN Inference," in *Proceedings of the 26th Asia and South Pacific Design Automation Conference (ASP-DAC '21)*, 2021. [Online]. Available: https://doi.org/10.1145/3394885.3431535

[68] T. Komoda, S. Hayashi, T. Nakada, S. Miwa, and H. Nakamura, "Power capping of CPU-GPU heterogeneous systems through coordinating DVFS and task mapping," in *Proceedings of the IEEE 31st International Conference on Computer Design (ICCD '13)*, 2013. [Online]. Available: https://doi.org/10.1109/ICCD.2013.6657064

[69] T. Tambe, C. Hooper, L. Pentecost, T. Jia, E.-Y. Yang, M. Donato, V. Sanh, P. Whatmough, A. M. Rush, D. Brooks, and G.-Y. Wei, "EdgeBERT: Sentence-Level Energy Optimizations for Latency-Aware Multi-Task NLP Inference," in *Proceedings of the 54th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO '21)*, 2021. [Online]. Available: https://doi.org/10.1145/3466752.3480095

[70] A. K. Kakolyris, D. Masouros, S. Xydis, and D. Soudris, "SLO-aware GPU DVFS for Energy-efficient LLM Inference Serving," *IEEE Computer Architecture Letters*, 2024. [Online]. Available: https://doi.org/10.1109/LCA.2024.3406038

[71] P. Zou, A. Li, K. Barker, and R. Ge, "Indicator-Directed Dynamic Power Management for Iterative Workloads on GPU-Accelerated Systems," in *Proceedings of the 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID '20)*, 2020. [Online]. Available: https://doi.org/10.1109/CCGrid49817.2020.00-37

[72] Y. G. Kim and C.-J. Wu, "AutoScale: Energy Efficiency Optimization for Stochastic Edge Inference Using Reinforcement Learning," in *Proceedings of the 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO '20)*, 2020. [Online]. Available: https://doi.org/10.1109/MICRO50266.2020.00090

[73] A. Jahanshahi, M. Rezvani, and D. Wong, "WattWiser: Power & Resource-Efficient Scheduling for Multi-Model Multi-GPU Inference Servers," in *Proceedings of the 14th International Green and Sustainable Computing Conference (IGSC '23)*, 2023. [Online]. Available: https://doi.org/10.1145/3634769.3634807