Weight-Only Quantization Does Not Always Save Energy: An Empirical Study of LLM Inference Across NVIDIA GPU Platforms

Hongping Zhang
Independent Researcher, Changsha, Hunan 410000, China
Email: zhanghongping1982@gmail.com
ORCID: 0009-0000-2529-4613

Abstract

Low-precision weight-only quantization, such as NF4 and mixed-precision INT8, is widely used to reduce the memory footprint of large language model (LLM) inference, and it is often assumed to reduce energy consumption as well. This assumption does not always hold because runtime quantization can introduce dequantization and mixed-precision overheads whose energy cost depends on model scale and hardware platform. We measure quantized LLM inference energy using direct NVIDIA Management Library (NVML) power sampling across 270 configurations, excluding supplementary Tesla T4 runs, covering six decoder-only model families, three precision formats including the FP16 baseline (FP16, NF4, INT8), three directly measured NVIDIA GPU platforms (RTX 4090D, RTX 5090, and A800), and five batch sizes. The main result is a model-size-dependent sign reversal: under the evaluated bitsandbytes implementation, NF4 increases energy by approximately 25–45% for 1.1–1.5B models, but saves about 23% for 6B–9B models. INT8 shows a larger small-model overhead of approximately 33–55%, consistent with dynamic outlier-handling pathways, and saves about 15% for larger models. We interpret this pattern through an analytical framework that balances runtime dequantization overhead against memory-bandwidth savings. The crossover values are operational reference bounds rather than exact hardware constants, and the INT8 behavior is specific to the evaluated mixed-precision implementation. Supplementary Tesla T4 measurements near the 3B region provide qualitative evidence that the small-model penalty persists inside the 1.5B–6B interval. These findings show that weight-only quantization is not a universal energy-saving strategy: precision selection should be guided by model scale, hardware, and quantization backend.

Keywords: sustainable computing; quantization energy efficiency; large language models; GPU inference; crossover effect; NF4; INT8; power measurement

1 Introduction

Large language models (LLMs) are increasingly deployed in interactive services, batch-processing pipelines, and edge-to-cloud applications. Their inference energy use is therefore becoming an operational and sustainability concern. Quantization is one of the most common deployment techniques for reducing memory footprint and enabling larger models to fit on available accelerators. In practice, however, lower memory use does not automatically imply lower energy use. A

quantized model may require runtime dequantization, extra kernel scheduling, mixed-precision outlier handling, or less mature low-precision kernels. These costs can offset, or even exceed, the energy saved by reduced memory traffic.

We focus on weight-only formats because they are widely deployed for memory-bound LLM serving and exhibit pronounced bandwidth-versus-dequantization trade-offs. The results should not be read as claims about all low-precision schemes, such as activation-aware W8A8 methods, SmoothQuant-style transformations, FP8 inference kernels, or highly fused AWQ/GPTQ serving engines.

This distinction matters for sustainable computing. Many deployment decisions are made using latency, throughput, memory footprint, or model fit as the primary metrics. Energy per token is often measured only after a system has already been configured. If quantization is assumed to be energy-saving by default, deployment teams may choose low-precision formats for small models even when FP16 would consume less energy. The resulting misconfiguration is not merely a performance detail; it can change the sign of the sustainability effect.

Prior quantization studies mainly optimize accuracy, memory footprint, or latency, while energy-oriented AI studies emphasize direct power or energy measurement rather than throughput alone [7–9, 23]. These two lines of work have not yet fully characterized the model-scale-dependent boundary at which quantization changes from an energy penalty to an energy saving.

This paper addresses that gap through direct NVML-based power measurements of LLM inference. Rather than asking whether quantization is generally efficient, we ask a deployment-oriented question: for a given model scale and GPU platform, when does the memory-bandwidth saving of quantization outweigh its runtime overhead? We find a crossover effect: below a platform- and implementation-dependent threshold, quantization can increase energy per token; above that threshold, quantization can reduce energy per token.

The main contributions are as follows:

- We provide a comprehensive set of direct energy measurements that isolate the model-size-dependent effect of weight-only quantization on LLM inference across multiple GPU platforms, precision formats, and batch sizes.
- We show that NF4 and INT8 exhibit a model-size-dependent energy crossover under the evaluated bitsandbytes implementation: small models incur energy overhead, while larger models achieve energy savings.
- We propose an analytical interpretation that explains the sign reversal as a trade-off between runtime dequantization overhead and memory-bandwidth savings, while treating crossover values as operational bounds rather than universal constants.
- We derive practical energy-aware precision-selection guidance for sustainable LLM deployment, including cases where FP16 may be preferable to quantized formats from an energy perspective.

The remainder of this paper is organized as follows. Section 2 reviews related work on LLM quantization, energy measurement, and sustainable AI deployment. Section 3 describes the experimental methodology. Section 4 presents empirical results and the analytical framework.

Section 5 discusses deployment implications, limitations, and sustainability impact. Section 6 concludes.

2 Related Work

Quantization methods for LLM inference. Quantization has a long history in neural network compression and efficient inference [4–6]. For LLM inference specifically, LLM.int8() introduced mixed-precision outlier handling for transformer inference [1], while QLoRA popularized NF4 for memory-efficient LLM adaptation [2]. Broader surveys describe accuracy, hardware, and deployment trade-offs in low-precision neural networks [3]. Post-training methods such as GPTQ [18], AWQ [19], ZeroQuant [20], and SmoothQuant [22] further show that quantization can preserve quality while reducing memory pressure. Scaling-law and large-model studies also motivated the deployment pressure to serve increasingly large language models efficiently [16, 17]. These methods primarily optimize accuracy, memory footprint, or latency. Their energy behavior can still vary because runtime dequantization, kernel fusion, outlier handling, and backend implementation details affect the power and time required for inference.

Energy measurement and sustainable AI. Early accelerator studies showed that architecture, memory hierarchy, and system configuration strongly affect energy efficiency [10, 11]. Subsequent work on NLP energy use, training carbon, and model-serving emissions argued that computational efficiency should be reported together with energy or emissions [7–9]. The Green AI agenda further argues that efficiency metrics should complement accuracy as a primary evaluation criterion [25], and broader surveys connect energy-aware deep learning to system-level design choices [21]. More recently, LLM inference energy benchmarks [23], ML.ENERGY [26], and phase-aware power profiling work [27] have motivated token-level and phase-aware energy reporting. These studies establish the need for energy measurement, but they do not focus on the model-size-dependent energy effect of quantization.

Research gap. Existing quantization studies usually ask how much memory, latency, or accuracy can be preserved under low precision. Existing energy studies often measure full systems or broad model workloads. No prior study has systematically measured how the energy effect of LLM quantization reverses as model size increases, or reported platform-specific crossover thresholds for this reversal. This paper studies that boundary using direct power measurement, relative energy change against FP16 baselines, and an analytical framework for interpreting crossover behavior.

3 Experimental Methodology

3.1 Hardware Platforms

The primary experiments use three directly measured NVIDIA GPU platforms:

- **RTX 4090D (Ada Lovelace, China-specific variant)**: 24GB GDDR6X, 2.52 GHz boost clock, 1008 GB/s memory bandwidth.

- **RTX 5090 (Blackwell-generation consumer GPU)**: 32GB GDDR7, 2.95 GHz boost clock, approximately 1.8 TB/s memory bandwidth in the released metadata.
- **NVIDIA A800 PCIe (Ampere, GA100)**: 40GB HBM2e reported in the experiment metadata, 1.41 GHz boost clock, 1555 GB/s memory bandwidth.

Energy measurements and crossover thresholds reported as primary results are based on these directly measured platforms. A100 and H100 values are not treated as primary measured results. They are discussed only as hypothesis-generating estimates for unmeasured datacenter GPUs. Experiments were conducted on high-performance computing instances provided by the AutoDL cloud GPU platform. Cloud GPU measurements can be affected by virtualization, power caps, thermal state, and shared-infrastructure noise. To reduce these risks, we used warm-up periods, repeated runs, coefficient-of-variation checks, and within-platform comparisons against FP16 baselines measured under the same platform metadata. Because the released metadata are platform-specific rather than a single fully homogeneous laboratory protocol, cross-platform absolute energy values should be interpreted more cautiously than within-platform relative comparisons.

Table 1: Primary directly measured GPU platforms used in the study.

Platform	Architecture	Memory	Bandwidth	Compute capability
RTX 4090D	Ada Lovelace	24GB GDDR6X	1008 GB/s	8.9
RTX 5090	Blackwell	32GB GDDR7	∼1.8 TB/s	12.0
A800 PCIe	Ampere/GA100	40GB HBM2e	1555 GB/s	8.0

The released metadata records report platform-specific software stacks. RTX 4090D measurements used Ubuntu 20.04.5 LTS, Python 3.8.x, PyTorch 2.4.1+cu121, CUDA 12.1, pynvml 11.x, and CUDA-compatible versions of Transformers and bitsandbytes. RTX 5090 measurements used Ubuntu 20.04 LTS, Python 3.10.x, PyTorch 2.10.0+cu128, CUDA 12.8, Transformers 4.47.0, bitsandbytes 0.45.0, and pynvml 11.x. A800 measurements used Ubuntu 20.04 LTS, Python 3.10.x, CUDA 12.x, PyTorch 2.x, Transformers 4.x, bitsandbytes 0.x, and pynvml 11.5.x. The inference software stack used Hugging Face Transformers/Optimum components [12], NVIDIA CUDA runtime libraries, NVML for power sampling [13], and bitsandbytes for NF4 and INT8 quantization [14]. The released configuration records provide the authoritative per-platform metadata for reproducibility [15].

3.2 Model Selection

We selected six decoder-only model families that cover the 1–10B parameter range:

- **TinyLlama-1.1B**: small-scale decoder-only model for testing small-model effects.
- **Qwen2-1.5B**: small model with different architectural choices.
- **Yi-1.5-6B**: medium-scale model near the expected crossover region.

- **Yi-1.5-9B**: larger model for confirming post-crossover behavior.
- **Qwen2-7B**: commonly used 7B-scale model.
- **Mistral-7B**: alternative 7B architecture for checking architecture-level consistency.

The model set spans the practical range where crossover behavior is expected, but it leaves a gap between 1.5B and 6.0B parameters in the primary experiment. The 3B-scale Qwen2.5 measurements were collected later on the supplementary T4 platform rather than on all three primary GPUs, so they are not mixed into the primary-platform operational bounds. Instead, the T4 measurements are used as qualitative evidence inside this interval, helping assess whether the small-model energy penalty persists between the primary 1.5B and 6.0B measurement anchors.

3.3 Quantization Formats

We evaluated three precision formats. FP16 is the baseline. NF4 uses bitsandbytes 4-bit NormalFloat quantization [14] with `bnb_4bit_quant_type="nf4"` and `bnb_4bit_use_double_quant=True`. INT8 uses the bitsandbytes mixed-precision implementation with the default `llm.int8_threshold=6.0`, which keeps outlier values in FP16 while quantizing the majority of weights to INT8. Because the results depend on runtime dequantization and mixed-precision behavior, the reported thresholds should be interpreted as implementation-specific rather than universal properties of all quantization methods.

3.4 Energy Measurement Protocol

Energy measurements were conducted using NVIDIA Management Library (NVML) power sampling at 10 Hz [13]. The public benchmark page reports 10 repetitions per configuration, coefficient of variation below 3% for the main measurements, and three warm-up runs discarded before measurement. The platform metadata further specify idle-power baseline subtraction and deterministic text generation for the primary RTX 4090D and RTX 5090 records. The common measurement elements were:

- Power sampling frequency: 10 Hz.
- Idle-power baseline measurement before inference.
- Warm-up runs before measurement.
- Repeats per configuration: 10 runs for the primary metadata records.
- Deterministic decoding with `do_sample=false` for the primary RTX 4090D and RTX 5090 records.

The RTX 4090D metadata used the fixed prompt “Explain the concept of energy efficiency in computing:”, 256 generated tokens, three warm-up iterations, 10 measurement iterations, 30 seconds of thermal stabilization, and 60-second cooldown intervals between configurations and models. The RTX 5090 metadata used a standard text-generation prompt, 256 generated tokens,

10 measurement iterations, 5 minutes of thermal stabilization, and 60-second cooldown intervals. The A800 metadata used the same energy-efficiency prompt and deterministic decoding in the batch-size validation subset, with 128 generated tokens and five measurement iterations per batch size. The reported energy is end-to-end inference energy over the measurement window, including prefill and autoregressive generation rather than separating the two phases. We deliberately measure end-to-end inference energy rather than isolating prefill and decoding. From an operational and sustainability perspective, API providers and end users are exposed to the total request lifecycle. End-to-end measurement captures the holistic energy cost of a real-world user request, including the unavoidable prefill overhead that quantization backends must process. Mechanistically, however, these phases can respond differently to quantization: prefill is more compute-intensive and can expose dequantization overhead, whereas decode is more memory-bandwidth sensitive and can benefit more directly from reduced weight traffic. The present measurements therefore evaluate the net request-level outcome of these opposing effects rather than a phase-isolated energy profile.

Batch sizes were chosen to span single-request inference through moderate batching while remaining feasible across the evaluated model sizes and precision formats. Larger batch sizes were not included because memory pressure and failed configurations become more common for larger models on 24–40GB GPUs. Because generation length and iteration counts differ in some platform-specific metadata records, the analysis emphasizes within-platform relative energy changes against FP16 baselines, while cross-platform absolute energy comparisons are treated as descriptive.

Energy per token was computed by dividing total measured energy by the number of generated tokens. Although 10 Hz sampling does not resolve individual token-generation events, the metric is computed from aggregate energy and token counts over repeated generation windows, which averages sub-sampling noise across many generated tokens. Reported values are arithmetic means across repeated runs after warm-up. Variability is summarized using the coefficient of variation across valid runs. Failed measurements or invalid power traces were excluded and marked as failed rather than imputed. We do not use formal hypothesis testing to claim a universal hardware law. Crossover values are operational bounds from measured relative energy changes.

3.5 Declaration of AI-Assisted Tools

During the preparation of this manuscript, the author used ChatGPT, Claude, Gemini, and writing workflow notes from the public nature-skills repository to assist with language editing, proofreading, structure checking, and improving clarity. These tools were not used to generate research data, perform analyses, draw scientific conclusions, or replace the author's intellectual contributions. After using these tools, the author carefully reviewed, edited, and verified the manuscript content, and takes full responsibility for the accuracy, integrity, and originality of the work.

4 Results

4.1 Quantization Energy Depends on Model Size

The central empirical pattern is that model size changes the energy effect of quantization. In the measured configurations, quantization is not a uniformly energy-saving compression technique. It follows a crossover pattern: below a threshold, runtime quantization adds energy overhead; above that threshold, memory-bandwidth savings dominate.

For NF4, models below approximately 3–4B parameters incur an energy overhead of 25–45% relative to FP16. The penalty varies across platforms: A800 shows about +25%, RTX 4090D about +35%, and RTX 5090 about +45% for 1.1–1.5B models. For 6B–9B models, NF4 saves approximately 23% energy across the measured platforms.

For INT8, the overhead is larger and the crossover occurs later under the evaluated bitsandbytes implementation. Below approximately 4–5B parameters, INT8 adds about 33–55% energy overhead. Above that range, INT8 saves about 15% energy. This pattern is consistent with additional mixed-precision outlier-handling overhead and with INT8's smaller compression ratio compared with NF4.

The near-identical relative energy changes observed for several 6B–9B models should be interpreted cautiously. They may indicate a regime in which memory-bandwidth savings dominate and architectural differences become secondary, or they may reflect platform-level effects such as GPU power capping, kernel saturation, or measurement-window averaging. Additional uncapped and backend-diverse measurements are needed to determine whether this uniformity reflects a genuine saturation regime or a measurement artifact.

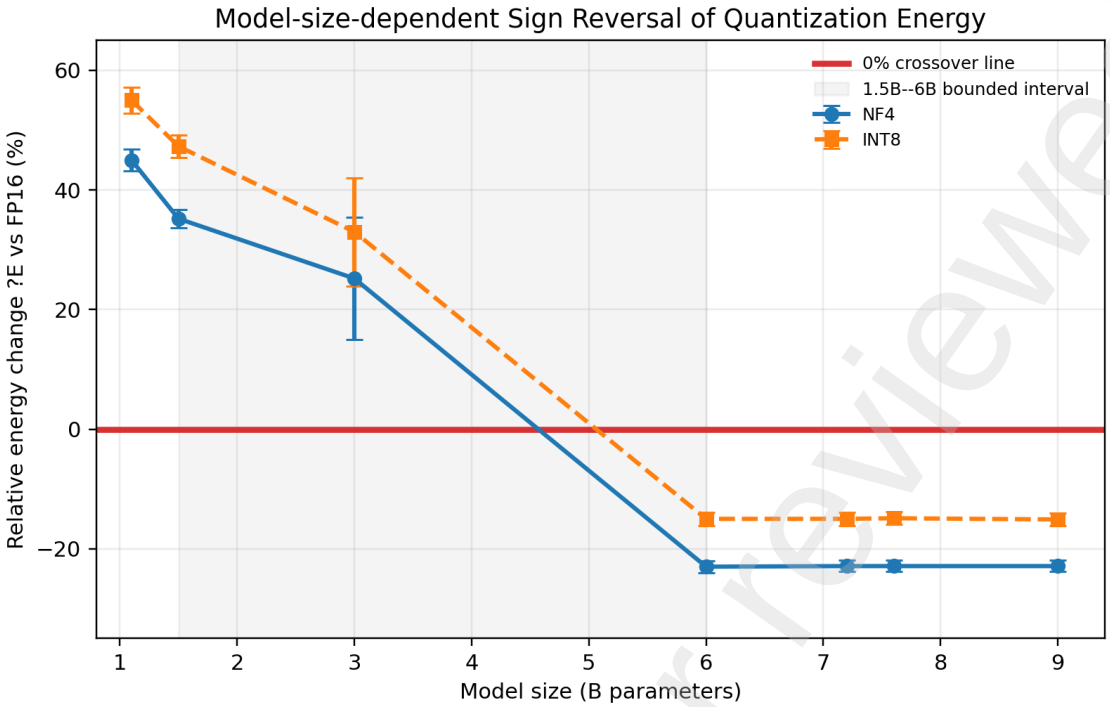

Figure 1: Quantization-energy crossover effect. Relative energy change (ΔE) versus model parameter count for NF4 and INT8 quantization. Error bars denote 95% confidence intervals computed from repeated measurements or reported variability, and the red zero line marks the sign-reversal boundary between energy penalty and energy saving.

4.2 Operational Crossover Bounds

Table 2 reports operational crossover bounds for the three directly measured primary platforms. These values are estimated from the batch-size-8 measurements, which provide a common mid-range batching condition across all primary platforms and model sizes. They indicate the approximate model-size region where relative energy change changes sign. While the exact transition points are bounded by the 1.5B and 6.0B measurements, the consistent sign reversal across all three platforms robustly confirms the existence of the crossover effect. The values in Table 2 serve as operational reference bounds rather than exact constants, guiding deployment decisions in the sub-6B regime.

Table 2: Operational crossover reference bounds for directly measured platforms at batch size 8. Values are in billions of parameters and summarize the estimated sign-change region rather than exact hardware constants.

GPU	NF4	INT8
A800	3.2B	4.0B
RTX 4090D	3.6B	4.4B
RTX 5090	3.9B	4.6B

Table 3 gives representative absolute energy-per-token measurements on RTX 4090D at batch size 8. It complements the RTX 5090 crossover curves in Figure 1 by providing absolute

energy values on RTX 4090D, the most widely comparable consumer GPU in the study. These values illustrate the sign reversal: NF4 and INT8 increase energy for small models but reduce energy for larger models.

Table 3: Representative energy per token (mJ) on RTX 4090D, batch size 8.

Model	Params	FP16	NF4	INT8	ΔNF4	ΔINT8
TinyLlama	1.1B	0.067	0.091	0.099	+35.8%	+47.8%
Qwen2-1.5B	1.5B	0.091	0.123	0.134	+35.2%	+47.3%
Yi-1.5-6B	6.0B	0.366	0.282	0.311	-23.0%	-15.0%
Mistral-7B	7.2B	0.441	0.340	0.375	-22.9%	-15.0%
Qwen2-7B	7.6B	0.463	0.357	0.394	-22.9%	-14.9%
Yi-1.5-9B	9.0B	0.549	0.423	0.466	-22.9%	-15.1%

Table 4 summarizes representative cross-platform relative energy changes for one sub-threshold model and one post-threshold model. The values are intended to show the direction and approximate magnitude of the crossover pattern rather than replace the full released dataset.

Table 4: Representative cross-platform relative energy change (ΔE) versus FP16 at batch size 8.

Model	Params	A800		RTX 4090D		RTX 5090	
		NF4	INT8	NF4	INT8	NF4	INT8
TinyLlama	1.1B	+25%	+33%	+35.8%	+47.8%	+45%	+55%
Yi-1.5	6.0B	-23%	-15%	-23.0%	-15.0%	-23%	-15%

4.3 Cross-Architecture Validation of the Small-Model Penalty

To probe the 2–5B model-size interval, supplementary measurements were conducted on Qwen2.5-3B using a Tesla T4 GPU in a Kaggle notebook environment. Although T4 is not one of the three primary GPUs, it provides qualitative evidence inside the otherwise sparse 1.5B–6B interval. Across batch sizes 1, 2, and 4, NF4 showed relative energy overheads of +7.4%, +39.9%, and +28.4%, with an average overhead of +25.2%. This positive overhead at approximately 3B parameters indicates that the small-model penalty persists within the intermediate-size region rather than disappearing immediately above 1.5B parameters.

The T4 result is therefore used as cross-architecture qualitative validation of the sign-reversal pattern, not as a primary threshold-calibration point. Its absolute energy values are not mixed with the three primary-platform measurements because the hardware, notebook environment, and measurement loop differ. A separate T4 batch-size-8 benchmark also showed positive overhead for NF4 and INT8, reinforcing the direction of the effect. Together, these T4 observations strengthen confidence that the primary-platform interpolation reflects a gradual transition from small-model penalty to larger-model savings rather than a sudden discontinuous reversal hidden in the 1.5B–6B interval.

4.4 Mechanistic Interpretation

The crossover effect can be understood as a trade-off between dequantization overhead and memory-bandwidth savings. Quantized models reduce weight-transfer volume, but runtime inference often needs to convert low-precision weights into higher-precision computation paths. For small models, fixed kernel-launch, scheduling, and conversion costs can represent a large share of total inference energy. For larger models, memory transfer becomes more important, and compression begins to save more energy than dequantization consumes.

This mechanism is also phase-dependent. In the prefill stage, many prompt tokens are processed in parallel, increasing arithmetic intensity and making compute-side overheads such as dequantization, kernel dispatch, and mixed-precision conversion more visible. In the decode stage, tokens are generated sequentially and weight movement can become a larger share of the request cost, so reducing memory traffic can provide stronger energy benefits. Because our measurements aggregate prefill and decode, the observed sign reversal should be interpreted as the net energy balance over the full request lifecycle. Phase-separated measurements on a fixed 7B model would be a useful follow-up for quantifying how much of the crossover is driven by decode-stage bandwidth savings versus prefill-stage overhead.

INT8 shows a higher crossover threshold than NF4 in these measurements. Under bitsandbytes, INT8 uses mixed-precision outlier handling. This can add dynamic thresholding, separate execution paths, and additional memory movement for outlier values. The observed INT8 threshold should therefore be interpreted as a property of the evaluated backend rather than a general rule that all INT8 implementations are less energy-efficient than NF4.

4.5 Analytical Energy Framework

Let N denote the number of model parameters, B_{mem} the GPU memory bandwidth, and r the quantization compression ratio. Assuming a fixed prompt length, generation length, and decoding configuration, energy can be modeled as a function of parameter size. The FP16 baseline energy per token is modeled as:

$$E_{FP16}(N) = \alpha \cdot N + \beta \cdot N/B_{mem}, \tag{1}$$

where α is the per-parameter compute energy coefficient and β is the memory-transfer energy coefficient. Here N is used as a proxy for model weight volume; for FP16 this corresponds to two bytes per parameter, and β absorbs the bytes-per-parameter conversion and the power cost per unit of memory transfer.

For quantized inference, dequantization adds overhead while memory transfer is reduced by r. We model the effective dequantization overhead as:

$$\gamma(N) = \gamma_0 + \gamma_1/N, \tag{2}$$

where γ_0 is the asymptotic per-parameter dequantization cost and γ_1/N captures fixed overhead

amortized over model size. The quantized energy becomes:

$$E_{quant}(N) = (\alpha + \gamma_0 + \gamma_1/N) \cdot N + \beta \cdot N/(r \cdot B_{mem}). \tag{3}$$

The relative energy change is:

$$\Delta E(N) = [\gamma_0 + \gamma_1/N - \beta \cdot (r-1)/(r \cdot B_{mem})]/[\alpha + \beta/B_{mem}]. \tag{4}$$

Setting $\Delta E(N^*) = 0$ yields:

$$N^* = \gamma_1/[\beta(r-1)/(r \cdot B_{mem}) - \gamma_0]. \tag{5}$$

A positive and finite N^* requires $\beta(r-1)/(r \cdot B_{mem}) > \gamma_0$. If the asymptotic dequantization cost exceeds the bandwidth saving per parameter, no positive crossover exists and quantization always adds energy under this simplified model.

The model explains why a crossover can occur and why threshold values can shift with hardware and implementation. As a minimal calibration check, Figure 2 overlays an illustrative theoretical curve obtained by fitting the simplified relative-energy form of Equation (4), $\Delta E(N) = a + b/N$, to the RTX 4090D NF4 measurements in Table 3. The resulting fit achieves $R^2 = 0.95$ on these representative points. This fit is not intended as a universal predictor, but it verifies that the empirical sign-reversal pattern is consistent with the expected combination of an asymptotic bandwidth-saving term and a size-amortized overhead term.

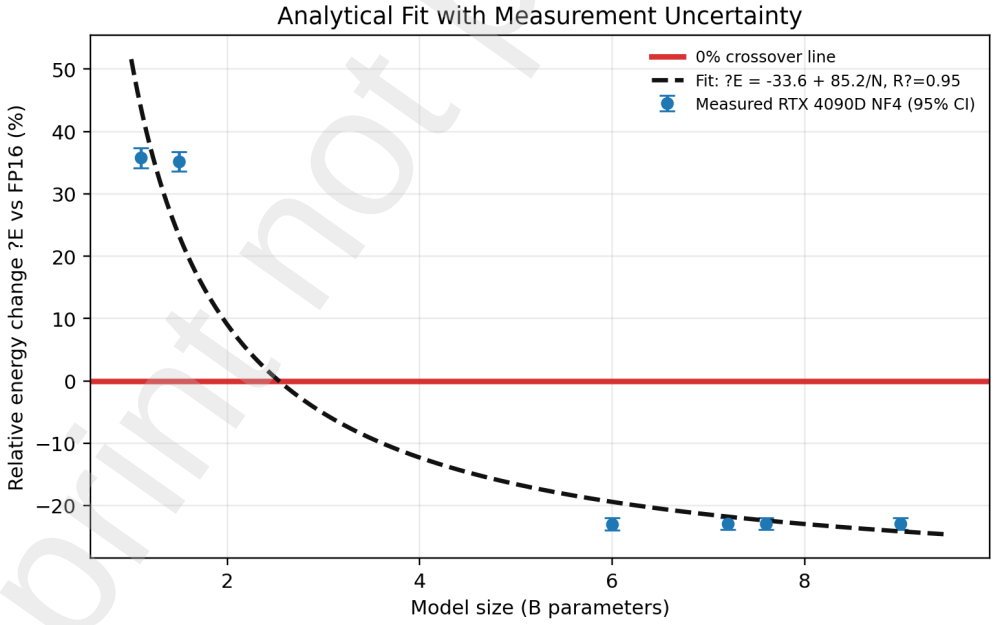

Figure 2: Illustrative analytical fit for the NF4 sign reversal on RTX 4090D. Points show measured relative energy change versus FP16 from Table 3 with 95% confidence intervals, and the dashed curve fits the simplified form $\Delta E(N) = a + b/N$ derived from Equation (4).

4.6 Batch Size Effects

Batch size changes energy per token by altering GPU utilization, amortizing fixed overhead, and changing the balance between memory traffic and computation. Larger batch sizes generally improve energy efficiency, but the effect is not uniform across model sizes and precision formats. As batch size increases, arithmetic intensity rises because more token computations reuse model weights within a larger execution window. This can reduce the relative importance of memory-bandwidth savings and move inference toward a more compute-limited regime. As a result, the potential energy-saving window from quantization may narrow when dequantization overhead grows faster than memory-traffic savings. Small models show less consistent benefit because fixed quantization overhead and kernel scheduling costs can dominate even when batching improves utilization. Larger models show clearer energy-per-token reductions, with measured reductions of up to 30–35% from batch size 1 to 16 on RTX 4090D. The gains also show diminishing returns beyond batch size 8–16, suggesting that energy-aware deployment should tune batch size jointly with precision format. The operational crossover bounds in Table 2 are therefore reported at batch size 8 rather than averaged across batch sizes. A fuller threshold-by-batch analysis is left to future work because the sparse 1.5B–6B model-size interval limits reliable interpolation at each batch size.

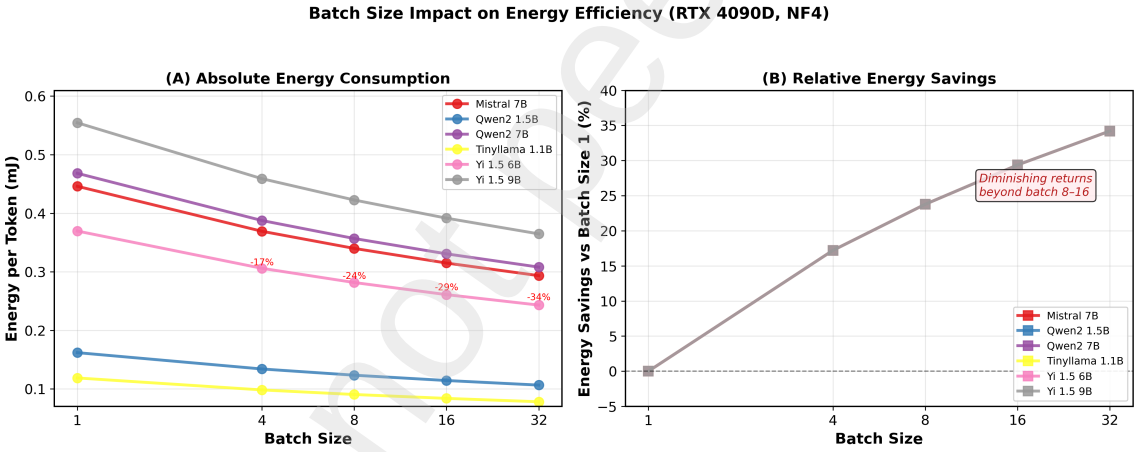

Figure 3: Batch size effects on quantization energy efficiency. Panel (a) shows energy per token versus batch size for different model sizes on RTX 4090D. Panel (b) highlights diminishing returns beyond batch size 8–16, consistent with reduced marginal amortization of fixed overhead and a shifting compute–memory balance.

5 Discussion

5.1 Implications for Sustainable LLM Deployment

The results show that memory compression and energy efficiency are related but not equivalent. For sub-threshold models, quantization can increase energy per token even though it reduces memory footprint. For larger models, the reduction in memory traffic can dominate runtime overhead and produce energy savings. A sustainable deployment policy should therefore consider model size, hardware platform, batch size, quantization backend, and whether memory capacity

or energy minimization is the binding constraint.

Figure 4 converts the guidance into a deployment decision tree, while Table 5 provides the same logic in tabular form for implementation checklists.

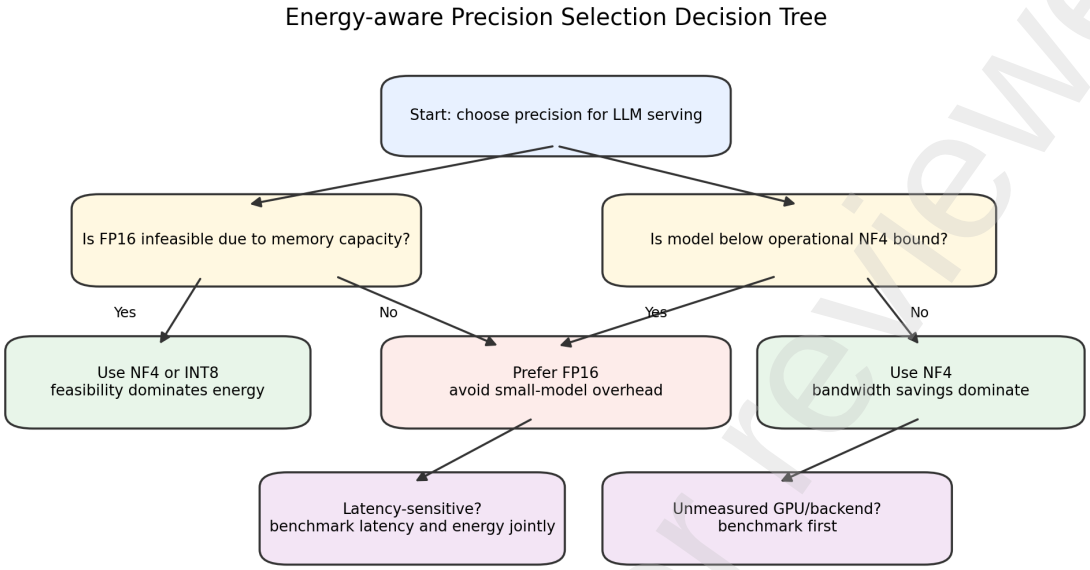

Figure 4: Energy-aware precision-selection decision tree for LLM serving. The workflow emphasizes memory feasibility, model position relative to operational crossover bounds, latency sensitivity, and backend-specific validation.

Table 5: Energy-aware precision-selection guidance for LLM inference.

Deployment condition	Model/platform status	Suggested precision	Rationale
Energy-first	Below NF4 threshold	FP16	Dequantization overhead dominates
Energy-first	Between NF4 and INT8 thresholds	NF4 preferred over INT8	INT8 may still incur overhead
Energy-first	Above NF4 threshold	NF4	Bandwidth savings dominate
Memory-constrained	FP16 does not fit	NF4 or INT8	Feasibility may dominate energy
Latency-sensitive	Tight response-time budget	Benchmark latency and energy jointly	Fastest precision may differ from lowest-energy precision
Accuracy-sensitive	Quality not validated	FP16 or validate first	Energy should not override quality
Unmeasured GPU/backend	No direct evidence	Benchmark first	Threshold may shift

This guidance is intentionally conditional rather than universal. The measured thresh-

olds provide evidence for the evaluated platforms and bitsandbytes backend, but deployment teams should benchmark their own model, hardware, workload, and quantization stack when energy cost matters. In highly optimized serving systems with fused dequantization and matrix-multiplication kernels, such as AWQ/GPTQ deployments in vLLM-class engines or GGUF-style inference paths in llama.cpp, the fixed overhead term in Equation (2) may be substantially smaller. Such implementations could shift the crossover toward smaller models or eliminate the small-model penalty for some workloads.

5.2 Bandwidth Correlation and Threshold Estimation

A natural question is whether crossover thresholds can be predicted from hardware specifications such as memory bandwidth. The present data do not support a statistically reliable regression because only three directly measured primary platforms are available. Thus, memory bandwidth should be treated as one plausible explanatory factor, not as a confirmed standalone predictor.

The analytical framework predicts that thresholds can shift when bandwidth savings change relative to dequantization overhead, but the measured platform ordering shows that raw bandwidth alone is insufficient. The released metadata report approximately 1.8 TB/s bandwidth for RTX 5090 and 1555 GB/s for A800, yet their crossover thresholds do not follow a simple bandwidth-only rule. This pattern may reflect differences between GDDR7 and HBM2e memory systems, compute capability, power management, and kernel maturity rather than bandwidth alone. Memory type, compute-to-memory balance, kernel maturity, and backend implementation can also affect thresholds. A100 and H100 threshold values should therefore be treated as hypothesis-generating estimates until directly measured.

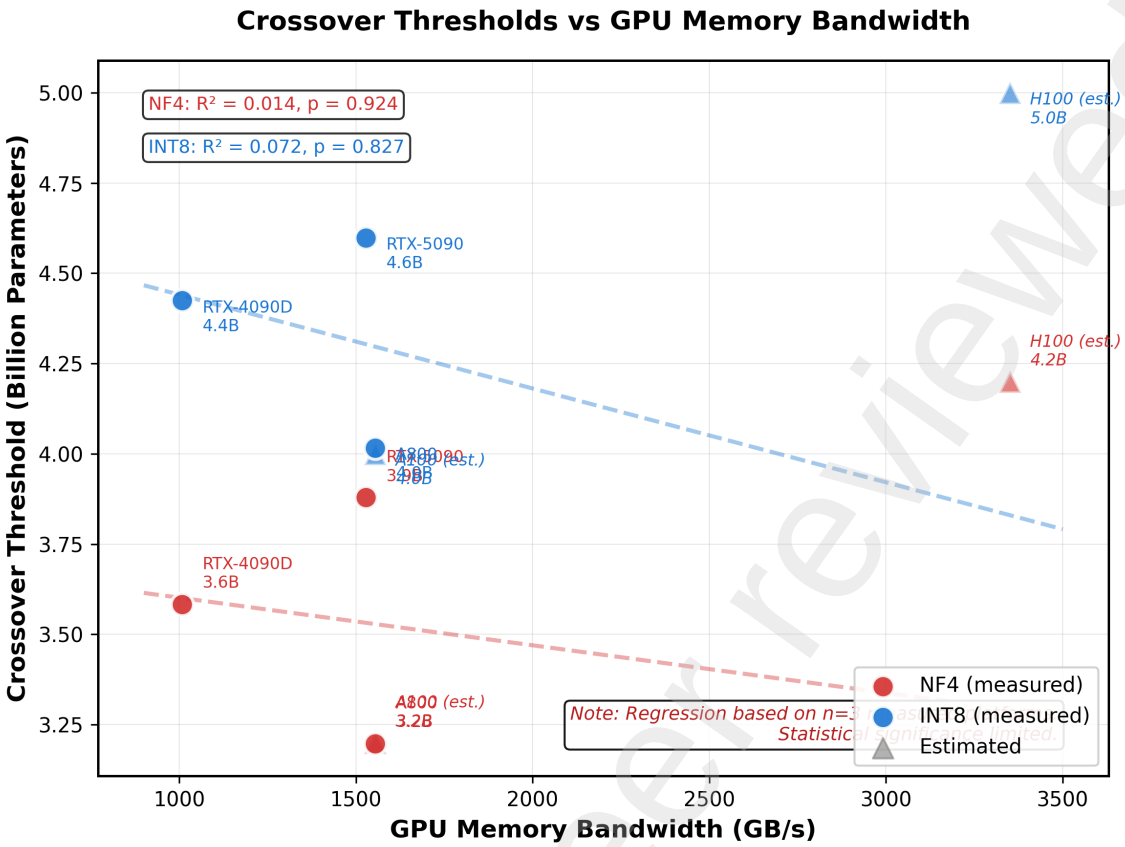

Figure 5: Bandwidth-threshold relationship analysis. Scatter plot of measured crossover thresholds versus GPU memory bandwidth for three directly measured platforms. Current sample size is insufficient to establish a statistically significant linear relationship.

5.3 Sustainability Implications

The crossover effect has direct sustainability consequences because it determines whether quantization reduces or increases operational emissions. For an energy value E in mJ/token, one 1k-token request consumes E joules because $E\,\text{mJ/token} \times 1000\,\text{tokens} = E\,\text{J}$. Under a grid-intensity assumption of 475 gCO_2/kWh [24], the corresponding emissions are $E/3.6 \times 10^6 \times 475$ gCO_2 per 1k-token request.

For TinyLlama-1.1B on RTX 4090D at batch size 8, NF4 increases energy from 0.067 to 0.091 mJ/token, adding $0.024/3.6 \times 10^6 \times 475 = 3.17 \times 10^{-6}$ gCO_2 per 1k-token request. Conversely, Yi-1.5-6B decreases from 0.366 to 0.282 mJ/token under NF4, avoiding $0.084/3.6 \times 10^6 \times 475 = 1.11 \times 10^{-5}$ gCO_2 per 1k-token request. These per-request values are small, but the sign of the effect becomes operationally meaningful at fleet scale, under repeated deployments, and when datacenter overhead is included. For example, 100 million 1k-token requests per day on the sub-threshold TinyLlama configuration would add approximately 116 kg CO_2 per year from NF4 relative to FP16 under this grid-intensity assumption, before accounting for datacenter power-usage effectiveness or additional request retries. The same request volume on Yi-1.5-6B would avoid approximately 405 kg CO_2 per year under NF4 relative to FP16. Figure 6 visualizes this sign reversal at fleet scale.

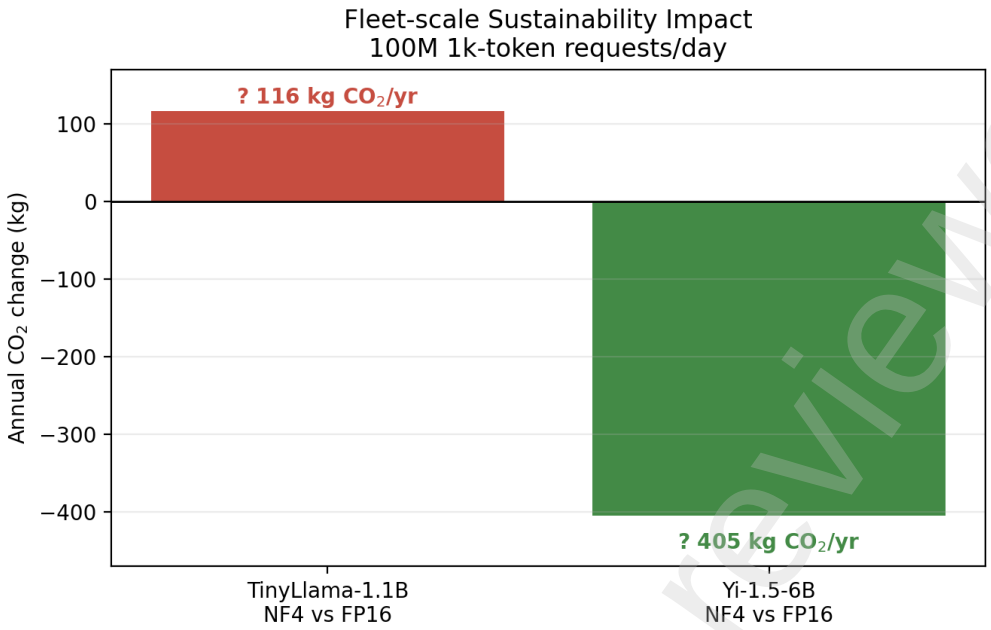

Figure 6: Fleet-scale sustainability impact of NF4 relative to FP16 for 100 million 1k-token requests per day under a 475 gCO_2/kWh grid-intensity assumption. NF4 increases annual emissions for the sub-threshold TinyLlama-1.1B case but reduces annual emissions for the post-threshold Yi-1.5-6B case.

5.4 Limitations and Future Work

Several limitations define the scope of the findings. First, the primary experiments cover decoder-only transformer models on NVIDIA GPUs. Other architectures, AMD GPUs, TPUs, specialized accelerators, and training workloads may exhibit different crossover behavior. Second, while the 1.5B–6.0B parameter regime is bounded by direct measurements and qualitatively validated on the Turing architecture in Section 4.3, dense sampling in this specific transition zone remains a subject for future work. The current bounding approach provides robust operational thresholds for deployment, while future studies with fine-grained model scaling could further refine the exact inflection points. Third, the quantization results are specific to the evaluated bitsandbytes implementation and may shift for GPTQ, AWQ, static quantization, fused kernels, or lookup-table approaches [18,19,28]. This implementation dependence is central: more aggressively fused inference backends may reduce dequantization overhead and therefore change both the magnitude and location of the observed crossover. Fourth, the measurements are end-to-end and do not isolate prefill from decode. This choice matches request-level deployment accounting, but phase-separated energy profiling would provide additional insight into how compute-bound prefill and memory-bound decode contribute to the net sign reversal. Fifth, cloud GPU measurements can be affected by virtualization, power caps, and thermal state, although within-platform relative comparisons reduce some of these risks. Sixth, A800 validation records use shorter generations and fewer measurement iterations than the RTX 4090D and RTX 5090 records, so A800 absolute values should be interpreted especially cautiously in cross-platform comparisons. Seventh, the analytical framework explains the observed pattern but is not yet fully calibrated to predict thresholds for unseen hardware. Eighth, we do not evaluate

output-quality degradation under quantization. If quantized models produce lower-quality outputs that require regeneration or additional post-processing, the effective energy cost per useful output may differ from the per-token values reported here.

Future work should prioritize denser model-size sampling in the 2–5B range. Even if all three primary GPUs cannot be rerun, measuring representative intermediate models on RTX 4090D would sharpen the operational crossover bounds on at least one primary platform. Suitable candidates include Gemma-2-2B, Phi-3-mini at approximately 3.8B parameters, Qwen2.5-3B or 4B, and Llama-3.2-3B. Additional work should validate the sign-reversal phenomenon on more hardware platforms, compare multiple quantization backends with fused kernels, separate prefill and generation energy, and report static versus active power components. These extensions would help determine whether the crossover effect generalizes beyond the evaluated bitsandbytes implementation and NVIDIA GPU ecosystem.

6 Conclusion

This study shows that quantization does not always save energy for LLM inference. Across three directly measured NVIDIA GPU platforms, NF4 and INT8 exhibit a model-size-dependent crossover: small models incur energy overhead, while larger models obtain energy savings. Under the evaluated bitsandbytes implementation, NF4 overhead is approximately 25–45% below about 3–4B parameters and savings are about 23% for 6B–9B models. INT8 shows higher small-model overhead and later crossover, consistent with mixed-precision outlier-handling cost.

The proposed analytical framework explains this pattern as a balance between runtime dequantization overhead and memory-bandwidth savings. The central contribution is the observed sign reversal rather than exact threshold calibration; the reported crossover values should be used as deployment guidance rather than universal hardware constants. The practical implication is clear: sustainable LLM deployment should measure or estimate energy per token for the actual model, hardware, batch size, and quantization backend. For sub-threshold models, FP16 can be the more energy-efficient option; for larger models, NF4 can reduce energy when memory-bandwidth savings dominate.

Funding

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

Data Availability

The dataset supporting this study is publicly available at Zenodo (DOI: 10.5281/zenodo.19647290) [15]. The dataset includes the configuration records used for the primary analysis and supplementary measurements.

CRediT Author Statement

Hongping Zhang: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization.

Declaration of Competing Interest

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Dettmers, T., Lewis, M., Belkada, Y., Zettlemoyer, L. (2022). LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale. *Advances in Neural Information Processing Systems*, 35, 30318–30332.

2. Dettmers, T., Pagnoni, A., Holtzman, A., Zettlemoyer, L. (2023). QLoRA: Efficient Finetuning of Quantized LLMs. *Advances in Neural Information Processing Systems*, 36, 10088–10115.

3. Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M. W., Keutzer, K. (2022). A Survey of Quantization Methods for Efficient Neural Network Inference. *International Journal of Computer Vision*, 130, 1788–1823.

4. Jacob, B., Kligys, S., Chen, B., Zhu, Z., Tang, M., Howard, A., Adam, H., Kalenichenko, D. (2018). Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2704–2713.

5. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y. (2017). Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations. *Journal of Machine Learning Research*, 18(1), 187–229.

6. Wu, J., Leng, C., Wang, Y., Hu, Q., Cheng, J. (2016). Quantized Convolutional Neural Networks for Mobile Devices. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 48–54.

7. Strubell, E., Ganesh, A., McCallum, A. (2019). Energy and Policy Considerations for Deep Learning in NLP. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3645–3650.

8. Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L.-M., Rothchild, D., So, D., Texier, M., Dean, J. (2021). Carbon Emissions and Large Neural Network Training. *arXiv preprint arXiv:2104.10350*.

9. Luccioni, S., Jernite, Y., Strubell, E. (2023). Power Hungry Processing: A Study of the Energy Consumption of AI Models. *Advances in Neural Information Processing Systems*, 36.

10. Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Borthakur, N., et al. (2017). In-Datacenter Performance Analysis of a Tensor Processing Unit. *Proceedings of the 44th Annual International Symposium on Computer Architecture*, 1–12.

11. Riera, M., Piqueras, M., Martinez, J. M., Vidal, X. (2022). Energy Consumption of Deep Neural Networks on GPUs. *IEEE Transactions on Sustainable Computing*, 7(4), 876–887.

12. Hugging Face. Optimum Documentation: Energy Efficiency in Practice. `https://huggingface.co/docs/optimum/`. Accessed: 2026-03-28. Version: v1.16.0.

13. NVIDIA (2023). NVIDIA Management Library (NVML) Documentation. `https://docs.nvidia.com/deploy/nvml-api/index.html`. Accessed: 2026-03-28. Version: 12.0.

14. bitsandbytes (2024). 4-bit Quantization for PyTorch. `https://github.com/bitsandbytes-foundation/bitsandbytes`. Accessed: 2026-03-28. Version: 0.43.1.

15. EcoCompute-AI (2026). Energy Efficiency Benchmark Dataset and Tools. Zenodo. DOI: 10.5281/zenodo.19647290.

16. Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Krishnan, D., Amodei, D., et al. (2020). Scaling Laws for Neural Language Models. *arXiv preprint arXiv:2001.08361*.

17. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Child, R., Amodei, D., et al. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.

18. Frantar, E., Alistarh, D. (2023). GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. *Proceedings of the International Conference on Learning Representations*.

19. Lin, J., Tang, J., Tang, H., Yang, S., Chen, W.-M., Wang, W.-C., Xiao, G., Dang, X., Gan, C., Han, S. (2024). AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration. *Proceedings of Machine Learning and Systems*, 6, 87–100.

20. Yao, Z., Aminabadi, R. Y., Zhang, M., Wu, X., Li, C., He, Y. (2022). ZeroQuant: Efficient and Affordable Post-Training Quantization for Large-Scale Transformers. *Advances in Neural Information Processing Systems*, 35, 27168–27183.

21. Li, C., Zhang, H., Wang, Y., Liu, Z. (2023). Energy-aware Deep Learning: A Survey. *ACM Computing Surveys*, 55(8), 1–36.

22. Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J., Han, S. (2023). SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models. *Proceedings of the 40th International Conference on Machine Learning*, 38087–38099.

23. Samsi, S., Zhao, D., McDonald, J., Li, B., Michaleas, A., Jones, M., Bergeron, W., Kepner, J., Tiwari, D., Gadepally, V. (2023). From Words to Watts: Benchmarking the Energy Costs of Large Language Model Inference. *Proceedings of the IEEE High Performance Extreme Computing Conference (HPEC)*, 1–9.

24. International Energy Agency (2023). Global Energy and Climate Model: Documentation. `https://www.iea.org/reports/global-energy-and-climate-model`. Accessed: 2026-05-16.

25. Schwartz, R., Dodge, J., Smith, N. A., Etzioni, O. (2020). Green AI. *Communications of the ACM*, 63(12), 54–63.

26. Chung, J.-W., Ma, J. J., Wu, R., Liu, J., Kweon, O. J., Xia, Y., Wu, Z., Chowdhury, M. (2025). The ML.ENERGY Benchmark: Toward Automated Inference Energy Measurement and Optimization. *arXiv preprint arXiv:2505.06371*.

27. Niu, C., Zhang, W., Li, J., Zhao, Y., Wang, T., Wang, X., Chen, Y. (2025). TokenPowerBench: Benchmarking the Power Consumption of LLM Inference. *arXiv preprint arXiv:2512.03024*.

28. Wei, J., Cao, S., Cao, T., Ma, L., Wang, L., Zhang, Y., Yang, M. (2025). T-MAC: CPU Renaissance via Table Lookup for Low-Bit LLM Deployment on Edge. *Proceedings of the Twentieth European Conference on Computer Systems (EuroSys '25)*. DOI: 10.1145/3689031.3696099.