The Cost of Context: Profiling the Energy Footprint of Input Tokens in Large Language Models

BORIS RUF, AXA AI Research, France
MARCIN DETYNIECKI, AXA AI Research, France

The rapid adoption of Retrieval-Augmented Generation (RAG) and long-context LLMs has shifted the computational burden toward the prefill phase—the initial stage where the model processes all input tokens in parallel before generating a response—yet its energy impact remains poorly characterised. We investigate the contradiction between massive hardware-level power spikes and the sub-linear marginal cost of input tokens reported in system-level studies. Through high-resolution telemetry of Llama-3.1 and Qwen-2 models on NVIDIA A100 hardware, we identify the efficiency threshold where $O(n^2)$ attention complexity overcomes GPU parallelisation. Our results quantify how optimised memory orchestration (PagedAttention) delays this scaling bottleneck across three orders of context scaling, yielding up to an 8.9× energy efficiency advantage at 112,000 tokens compared to standard baselines. Furthermore, we characterise the energy disparity between input and output tokens, demonstrating that optimised serving (vLLM) causes this ratio to narrow (from ≈850× to 750×) as context scales due to high decode efficiency, whereas unoptimised baselines show a widening intensity gap (reaching ≈220×) as $O(n)$ memory tax on decoding mounts. This work establishes context-scaling laws that differentiate engine-level optimisations from inherent model constraints, providing an empirical framework for sustainable LLM deployment.

CCS Concepts: • **Hardware → Power and energy**; • **Computing methodologies → Natural language processing**; *Machine learning*.

Additional Key Words and Phrases: Large Language Models, Energy Efficiency, Sustainability, PagedAttention, Prefill Phase

1 Introduction

The rapid proliferation of energy-intensive Large Language Models (LLMs) has made efficiency a key focus of systems research. Frontier models are predominantly accessible via APIs that prevent direct energy measurement, resulting in reliance on approximations [8]. Common estimation methodologies are based on output token counts, assuming that the primary driver of inference costs is autoregressive, sequential token generation, while input tokens are omitted [12].

In practice, however, the vast majority of modern LLM applications are heavily dependent on massive input contexts. Retrieval-Augmented Generation (RAG), long-document summarisation, and complex coding assistants all represent use cases where the input volume significantly outweighs the generated output.

This raises the question of how input tokens actually influence the energy balance, as recent research presents diverging results. Hardware-centric benchmarks like TokenPowerBench [10] report that inference engines such as vLLM [7]—a high-throughput LLM serving framework—spike GPU power draw by up to 3× during prefill. This aligns with the $O(n^2)$ complexity of self-attention [15], which arises because the mechanism requires pairwise comparisons between all tokens in the sequence, causing computational requirements to grow with the square of the context length. Conversely, systematic usage studies [2] observe that processing even very large contexts (e.g., 50,000 tokens) consumes only marginally more total energy than small prompts, suggesting a sub-linear marginal cost. This creates an apparent "Prefill Paradox" where high instantaneous power draw is seemingly decoupled from total energy expenditure.

This discrepancy highlights a lack of understanding of how context-length scaling affects energy costs in LLM inference, emphasising the need to characterise the prefill and decode phases independently.

In this short paper, we address the following research questions:

(1) At what context length threshold does the $O(n^2)$ complexity of attention overcome GPU parallelisation, and how does this point of compute saturation differ across engines?
(2) How do distinct memory-orchestration strategies (PagedAttention vs. SDPA) influence the location and stability of the efficiency threshold as context length scales?
(3) How does the energy intensity disparity between input and output tokens evolve across varying context depths, and how is this relationship modulated by engine-level optimisations?

2 Background

LLM inference proceeds in two distinct phases with fundamentally different computational profiles [11]. The *prefill* phase processes all input tokens in parallel: the model reads the entire prompt and computes a Key-Value (KV) cache that encodes the contextual relationships between tokens via the self-attention mechanism [15]. Because all tokens are processed simultaneously, this phase is *compute-bound*—it fully utilises the GPU's parallel arithmetic units. The *decode* phase then generates output tokens one at a time, where each new token attends to the full KV-cache built during prefill. This sequential dependency makes decoding *memory-bound*—the GPU spends most of its time loading model weights and cache entries rather than performing arithmetic [7].

This asymmetry has direct energy implications: prefill draws high instantaneous power (the GPU is fully active) but completes quickly, while decode draws lower power but runs for many sequential steps. Understanding how this balance shifts with context length is central to this work.

3 Related Work

Understanding LLM energy consumption requires bridging hardware telemetry with application-level metrics. TokenPowerBench [10] provides a foundation for measuring instantaneous power draw by isolating prefill and decode stages, revealing that energy per token rises faster than parameter count due to memory traffic penalties. Caravaca et al. [2] offer a longitudinal perspective, observing

Authors' Contact Information: Boris Ruf, boris.ruf@axa.com, AXA AI Research, Paris, France; Marcin Detyniecki, marcin.detyniecki@axa.com, AXA AI Research, Paris, France.

that while both input and output tokens contribute to energy footprints, output generation is significantly more intensive (e.g., a reported 11:1 energy disparity for output-heavy prompt configurations). This aligns with findings from the MELODI framework [6] that energy consumption is weakly correlated with prompt length (0.075) but strongly correlated with response length (0.846). Complementing these energy-centric studies, LLMCO2 [5] improves inference-time carbon-footprint prediction, helping connect measured energy behavior to emissions-aware reporting.

Recent modelling efforts have quantified these dynamics using polynomial formulations. Wilkins et al. [16] and Wu et al. [17] model inference energy as a function of T_{in} and T_{out}, highlighting a massive disparity where processing context for "memory formation" (e.g., RAG) can consume $1,000\times$ more energy than simple indexing. Nguyen et al. [9] further discuss system-level trade-offs in sustainable LLM serving, reinforcing that phase-specific efficiency must be evaluated together with deployment-level constraints. Furthermore, large-scale studies have diagnosed the underlying mechanisms of consumption, identifying system-wide bottlenecks such as memory pressure and KV-cache availability [3]. In multimodal regimes, the prefill cost for non-text inputs (images/video) can increase by up to $15\times$ compared to text-only baselines [3].

Optimisation techniques attempt to mitigate these costs through structural interventions. These include memory management through PagedAttention (*vLLM* [7]), phase-splitting (*Splitwise* [11]), chunked prefills (*Sarathi-Serve* [1]), and KV-cache materialization in flash storage (*MatKV* [13]). Additionally, the resource asymmetry between the prefill and decode phases has prompted new scheduling paradigms (*DistServe* [18]) to address the "Sustainable AI Trilemma" [17] of balancing AI capability, digital equity, and environmental sustainability. In the same spirit, Sinha et al. [14] caution against relying on utilization as a standalone sustainability proxy and emphasize mechanism-level attribution for inference-serving decisions.

4 Theoretical Analysis

We hypothesise that the quadratic complexity of self-attention eventually dominates energy consumption due to three fundamental physical hardware limits:

(1) **High Parallel Efficiency Phase:** For small contexts, GPUs maximise throughput by processing tokens in parallel. In this regime, the energy per token effectively decreases as the fixed startup and weight-loading costs are amortised over more input tokens.
(2) **The Compute Saturation Cliff:** Once the context size N exceeds the GPU's capacity for simultaneous parallel execution (limited by active hardware compute units), the engine must resort to sequential processing of attention chunks. At this point, the $O(n^2)$ complexity is no longer masked, and the energy tax becomes measurable.
(3) **The Memory Bandwidth Wall:** Calculating self-attention for massive sequences requires storing and frequently accessing the $N \times N$ attention probability matrix. As N grows, the volume of data movement between the VRAM and compute cores scales quadratically, eventually saturating the memory bus. Energy consumption then becomes dominated by the high Joule-per-bit cost of off-chip data transfers.

5 Experimental Design

To investigate the energy impact of massive inputs, we implement a modular benchmarking suite designed to bridge the gap between coarse application-level estimates and fine-grained hardware telemetry. Our experimental design draws inspiration from the architecture of TokenPowerBench [10] and the feature-aware analysis of MELODI [6].

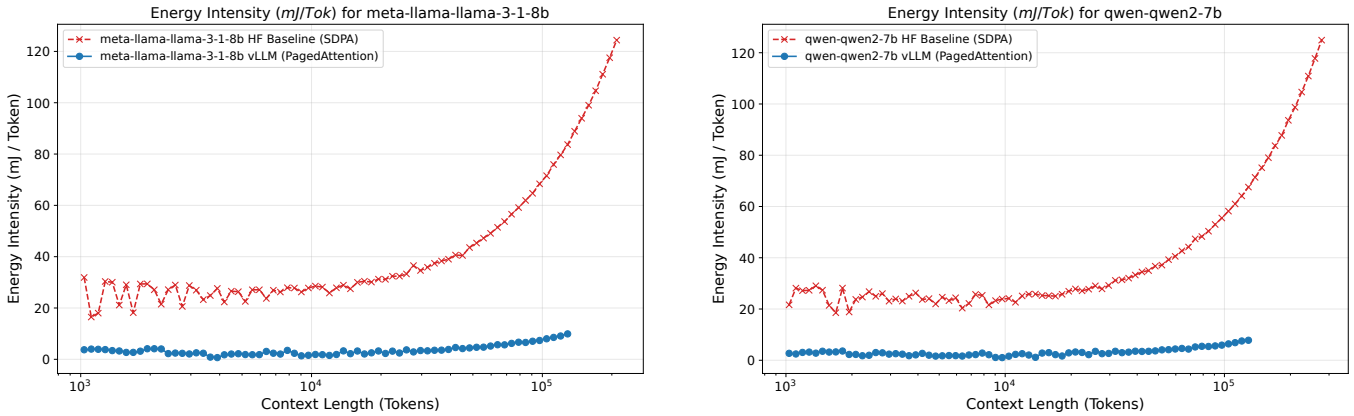
5.1 Instrumentation and Telemetry

We utilise a multi-threaded telemetry manager that samples GPU power draw via the NVIDIA Management Library (NVML) at a high-resolution 1ms sampling rate, consistent with NVML's documented minimum update interval for instantaneous power queries. While CPU/DRAM telemetry (e.g., RAPL) is often restricted in virtualised cloud environments, our methodology follows the hardware-centric approach of Caravaca et al. [2], focusing on the GPU as the primary energy consumer in LLM prefill phases. Each power sample is timestamp-aligned with the inference engine's event markers (prefill_start and prefill_end). In our benchmarks, the latter denotes the completion of the first-token generation, effectively isolating the context-processing phase. This allows us to integrate power over precisely defined intervals to calculate E_{prefill}.

5.2 Workload Configuration

To identify the exact efficiency threshold where $O(n^2)$ complexity breaks hardware parallelisation, we utilise a logarithmic context sweep. This approach provides the deterministic control necessary for high-resolution phase transition detection.

We generate synthetic prompts of length $L \in [2^8, 2^{17}]$ by repeating high-entropy base text and truncating to exact token counts. Since the attention mechanism performs identical matrix operations regardless of token content (i.e., the computational cost depends on sequence length, not semantic meaning), the use of repeated text does not affect the measured energy profile. We use model-specific tokenisers (e.g., Llama-3 128k and Qwen-2 1M) to ensure exact sequence lengths. Note that the maximum tested context length is bounded by each engine's supported context window; vLLM enforces the model's declared context limit (128k for Llama-3.1), whereas the HF Baseline permits extrapolation beyond this boundary, resulting in additional data points at higher context lengths in Figures 1 and 3. This granular control eliminates the noise of variable semantic content and ensures that the x-axis (Context Length) is perfectly uniform, allowing for the precise calculation of the discrete derivative of energy consumption $\frac{dE}{dL}$. The benchmarks are conducted on NVIDIA A100 (80GB) instances across two model families (Llama-3.1 8B, Qwen-2 7B) using bfloat16 precision to account for architectural variability in attention implementation. This configuration (A100 hardware, bfloat16 precision, model-native tokenisers, NVML-based telemetry) is consistent with the experimental setups used in TokenPowerBench [10] and MELODI [6].



(a) Llama-3.1 8B

(b) Qwen-2 7B

Fig. 1. Energy Intensity (mJ/Tok) across context lengths. Note the efficiency threshold in the HF baseline compared to the delayed efficiency inflection point in vLLM.

5.3 Controlled Inference Scenarios

We utilise two distinct inference engines as our experimental groups: (1) **Hugging Face Transformers** (hereafter **HF Baseline**), utilising the standard "Eager" implementation with Scaled Dot-Product Attention (SDPA) and contiguous KV-cache allocation; and (2) **vLLM (Optimised)** [7], which uses PagedAttention to optimise memory management and minimise fragmentation. In both cases, we isolate the prefill phase by setting the generation length to exactly 1 token. Each configuration is preceded by a warm-up pass to stabilise GPU clock speeds and power states. As our workload is fully deterministic (fixed synthetic inputs, greedy decoding, no batching), each context length is measured in a single run; prior work on similar deterministic GPU benchmarks has shown negligible run-to-run variance under these conditions [10].

5.4 Comparative Intensity Protocol

To characterise the energy disparity between input and output tokens, we conduct a secondary benchmarking run. In this configuration, we fix the output length to $G = 128$ tokens across the same logarithmic context sweep. By calculating the total energy E_{total} and subtracting the isolated prefill energy $E_{prefill}$ obtained in the previous step, we derive the mean energy per output token $E_{out} = (E_{total} - E_{prefill})/G$. This allows for a direct intensity comparison (E_{out}/E_{in}) across varying context depths.

6 Experimental Evaluation

Our evaluation characterises the energy-to-context scaling law across two model families (Llama-3.1 8B and Qwen-2 7B) on NVIDIA A100 (80GB) hardware. We analyse the resulting dataset through three primary lenses: (1) **Energy Intensity Scaling** to identify the hardware parallelisation threshold, (2) **Input vs. Output Proportion** to quantify energy disparity (E_{out}/E_{in}), and (3) **Component-Wise Attribution** to decompose the footprint using the correlation techniques introduced in MELODI [6]. By comparing the standard Hugging Face SDPA baseline against the optimised vLLM engine (PagedAttention), we observe a dramatic divergence in energy intensity as context length scales.

6.1 Energy Intensity Scaling

The experimental data reveals that while both engines exhibit quadratic scaling tendencies at extreme context lengths, the marginal cost of an input token ($\frac{dE}{dL}$) begins to rise significantly earlier in the baseline implementation. This leads to an efficiency disparity between optimised and standard serving that is nearly an order of magnitude (see Figure 1). As detailed in Table 1, at approximately 112,000 tokens, the vLLM engine processed the Llama-3.1 input in 3.54 seconds (1,005 Joules), while the baseline required 25.54 seconds (7,600 Joules). This represents a 7.6× improvement in energy efficiency for the optimised engine (reaching up to 8.9× for Qwen-2). These results confirm Hypothesis 1 (Section 4): at small contexts, both engines benefit from high parallel efficiency, but the threshold where Hypothesis 2 manifests differs by an order of magnitude between engines.

Table 1. Prefill Energy Intensity at $L \approx 112k$ tokens.

Model / Engine	Lat. (s)	Energy (J)	J/Tok
Llama-3.1 8B			
HF Baseline	25.54	7,600.2	0.0678
vLLM (Paged)	**3.54**	**1,005.1**	**0.0090**
Qwen-2 7B			
HF Baseline	23.04	6,832.0	0.0610
vLLM (Paged)	**2.71**	**769.6**	**0.0069**

6.2 Input vs. Output Proportion

To provide a complete picture of operational costs, we analyse the energy relationship between prefill (input) tokens and decode (output) tokens (see Figure 2). Output token energy is dominant in both regimes due to sequential dependencies, but the *magnitude* of this

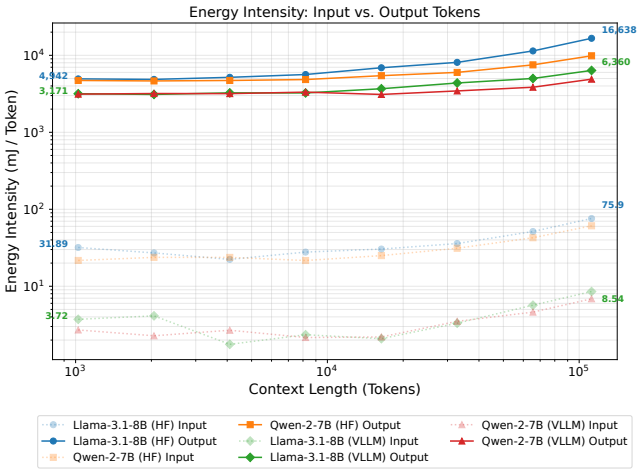


Fig. 2. Comparative energy intensities (mJ/Tok) for input (E_{in}) and output (E_{out}) tokens. For optimised engines, the input intensity remains flat, while unoptimised engines show E_{in} rising to meet E_{out}.

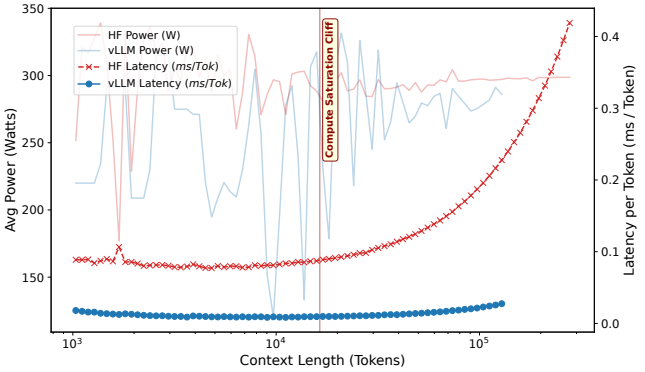


Fig. 3. The Compute Saturation Cliff: Phase Map of Power vs. Normalised Latency. While power draw saturates early at 310W, the latency per token begins to increase for unoptimised attention once hardware parallelisation is exceeded.

dominance varies dramatically between optimised and unoptimised engines.

At small context lengths ($L = 1k$), we observe that one output token consumes approximately 850× more energy than an input token in vLLM. Interestingly, as context scales to 112,000 tokens, this intensity disparity in optimised engines actually *decreases* to ≈750×. This is because vLLM's PagedAttention keeps the marginal cost of decoding low, while the $O(n^2)$ prefill math eventually increases the relative cost of input tokens.

In contrast, the HF Baseline ratio starts at ≈850× at small contexts but *widens* significantly, reaching ≈220× at 112k context. This is the hallmark of unoptimised environments, where the sequential memory tax on decoding grows much faster than the prefill costs, causing the energy cost of output tokens to explode relative to the parallel prefill phase.

6.3 Component-Wise Attribution

To investigate the drivers of the efficiency threshold, we analyse the relationship between instantaneous power draw (P_{avg}) and normalised latency (ms per token). Our telemetry indicates that for contexts exceeding 16,384 tokens, the GPU power draw plateaus at approximately 280W–310W, representing the thermal and electrical saturation of the A100—marking the entry into the **Compute Saturation Cliff** (see Figure 3).

In this saturation regime, the energy intensity is primarily driven by *latency extension* rather than increased power density. For the HF Baseline, we observe that normalised latency scales linearly ($O(n)$) with context length, implying that the total energy footprint is increasingly dominated by the quadratic ($O(n^2)$) total complexity of the **Memory Bandwidth Wall**. Specifically, the movement of massive attention matrices between VRAM and the compute cores accounts for the majority of the Joule-per-token increase.

In contrast, vLLM's use of kernel fusion (FlashAttention [4]) and efficient memory orchestration (PagedAttention [7]) significantly reduces memory traffic by keeping intermediate attention states in on-chip SRAM. Our results show that vLLM's normalised latency remains nearly flat until $L > 32k$, implying that it manages to sustain **High Parallel Efficiency** much longer than the baseline. This attribution confirms all three hypotheses from Section 4: the power plateau marks the Compute Saturation Cliff (Hypothesis 2), the linear latency growth reflects the Memory Bandwidth Wall (Hypothesis 3), and vLLM's ability to delay both transitions demonstrates sustained High Parallel Efficiency (Hypothesis 1).

7 Discussion

The empirical profiling of LLM prefill energy reveals that the apparent contradiction in prefill energy scaling is primarily an orchestration challenge rather than an inherent arithmetic limitation. Unoptimised baselines hit an efficiency threshold because they fail to mask the quadratic growth of self-attention memory traffic. In contrast, optimised engines like vLLM maintain a flat J/T profile across three orders of magnitude by aggressively amortising fixed hardware costs through parallel execution. This identifies the "Prefill Paradox" not as a contradiction in data, but as a manifestation of the "Race to Finish" principle: the high instantaneous power draw reported by hardware benchmarks is the precise mechanism that enables the sub-linear energy-per-token costs observed in system-level studies.

This divergence is explained by the fundamental difference between *arithmetic intensity* and *peak hardware utilisation*. Prefill's high arithmetic intensity enables the GPU to reach a state of maximum parallel efficiency, where power consumption per token is minimised despite high instantaneous power draw. The key implication is that, for modern optimised text-only stacks, input tokens remain effectively negligible relative to output tokens even at context lengths exceeding 100,000—though this balance may shift for multimodal inputs or future architectural changes.

8 Conclusion

By investigating the prefill energy contradiction, this research bridges the gap between hardware-level power measurements and application-level energy footprints. Our empirical results find that while GPU power draw saturates at ≈16k tokens, unoptimised baselines suffer from the $O(n^2)$ energy tax almost immediately once parallel overhead is exceeded. In contrast, optimised engines maintain an efficient linear J/T profile until $L > 32k$. Memory-orchestration strategies like PagedAttention and FlashAttention prove to be critical factors, delaying the efficiency threshold and yielding up to an 8.9× energy efficiency advantage at 112k tokens. Furthermore, we demonstrate how engine optimisations shift the intensity disparity between input and output tokens: optimised systems show a narrowing gap as they manage memory effectively, while unoptimised systems exhibit a widening gap reaching ≈220:1. This explains why the power spike of prefill is a signature of parallel efficiency, shifting the focus of sustainable AI from model constraints to infrastructure-level optimisations.

9 Future Work

Building on these findings, we identify three critical vectors for future energy-aware LLM research. First, *high-throughput amortisation* studies should investigate how request and continuous batching specifically compress the decode-phase energy disparity by further amortising weight-loading costs. Second, research into *precision-energy trade-offs* is required to evaluate whether the memory traffic reduction of FP8 or INT4 quantisation outweighs the potential power overhead of specialised low-precision tensor units. Finally, *architectural scalability* sweeps should be extended to Mixture-of-Experts (MoE) and 70B+ parameter models to determine if the scaling bottleneck shifts predictably with increased parameter count and activated memory footprints.

References

[1] Amey Agrawal, Nitin Kedia, Ashish Panwar, Jayashree Mohan, Nipun Kwatra, Bhargav S. Gulavani, Alexey Tumanov, and Ramachandran Ramjee. 2024. Taming throughput-latency tradeoff in LLM inference with sarathi-serve. In *Proceedings of the 18th USENIX Conference on Operating Systems Design and Implementation* (Santa Clara, CA, USA) *(OSDI'24)*. USENIX Association, USA, Article 7, 18 pages.

[2] Francisco Caravaca. 2026. Measuring Energy Consumption of LLMs Inferences. *SIGMETRICS Perform. Eval. Rev.* 53, 3 (Jan. 2026), 18–19. doi:10.1145/3788882.3788890

[3] Jae-Won Chung, Ruofan Wu, Jeff J. Ma, and Mosharaf Chowdhury. 2026. Where Do the Joules Go? Diagnosing Inference Energy Consumption. arXiv:2601.22076 [cs.LG]

[4] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FLASHATTENTION: fast and memory-efficient exact attention with IO-awareness. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '22)*. Curran Associates Inc., Red Hook, NY, USA, Article 1189, 16 pages.

[5] Zhenxiao Fu, Fan Chen, Shan Zhou, Haitong Li, and Lei Jiang. 2025. LLMCO2: Advancing Accurate Carbon Footprint Prediction for LLM Inferences. *SIGENERGY Energy Inform. Rev.* 5, 2 (Aug. 2025), 63–68. doi:10.1145/3757892.3757901

[6] Erik Johannes Husom, Arda Goknil, Lwin Khin Shar, and Sagar Sen. 2024. The Price of Prompting: Profiling Energy Use in Large Language Models Inference. arXiv:2407.16893 [cs.CY]

[7] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles* (Koblenz, Germany) *(SOSP '23)*. Association for Computing Machinery, New York, NY, USA, 611–626. doi:10.1145/3600006.3613165

[8] Sasha Luccioni, Yacine Jernite, and Emma Strubell. 2024. Power Hungry Processing: Watts Driving the Cost of AI Deployment?. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency* (Rio de Janeiro, Brazil) *(FAccT '24)*. Association for Computing Machinery, New York, NY, USA, 85–99. doi:10.1145/3630106.3658542

[9] Sophia Nguyen, Beihao Zhou, Yi Ding, and Sihang Liu. 2025. Towards Sustainable Large Language Model Serving. *SIGENERGY Energy Inform. Rev.* 4, 5 (April 2025), 134–140. doi:10.1145/3727200.3727220

[10] Chenxu Niu, Wei Zhang, Jie Li, Yongjian Zhao, Tongyang Wang, Xi Wang, and Yong Chen. 2026. TokenPowerBench: Benchmarking the power consumption of LLM inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 40. 32582–32590.

[11] Pratyush Patel, Esha Choukse, Chaojie Zhang, Aashaka Shah, Íñigo Goiri, Saeed Maleki, and Ricardo Bianchini. 2024. Splitwise: Efficient Generative LLM Inference Using Phase Splitting. In *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)*. 118–132. doi:10.1109/ISCA59077.2024.00019

[12] Samuel Rincé and Adrien Banse. 2025. EcoLogits: Evaluating the Environmental Impacts of Generative AI. *Journal of Open Source Software* 10, 111 (2025), 7471. doi:10.21105/joss.07471

[13] Kun-Woo Shin, Jay H. Park, Moonwook Oh, Yohan Jo, Jaeyoung Do, and Sang-Won Lee. 2025. MatKV: Trading Compute for Flash Storage in LLM Inference. arXiv:2512.22195 [cs.DC]

[14] Prasoon Sinha, Dimitrios Liakopoulos, Ruihao Li, and Neeraja J. Yadwadkar. 2025. The Utilization Fallacy and the Real Drivers of Carbon-Efficient Inference Serving. *SIGENERGY Energy Inform. Rev.* 5, 2 (Aug. 2025), 76–83. doi:10.1145/3757892.3757903

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6000–6010.

[16] Grant Wilkins, Srinivasan Keshav, and Richard Mortier. 2025. Offline Energy-Optimal LLM Serving: Workload-Based Energy Models for LLM Inference on Heterogeneous Systems. *SIGENERGY Energy Inform. Rev.* 4, 5 (April 2025), 113–119. doi:10.1145/3727200.3727217

[17] Hui Wu, Xiaoyang Wang, and Zhong Fan. 2025. Addressing the sustainable AI trilemma: a case study on LLM agents and RAG. arXiv:2501.08262 [cs.AI]

[18] Yinmin Zhong, Shengyu Liu, Junda Chen, Jianbo Hu, Yibo Zhu, Xuanzhe Liu, Xin Jin, and Hao Zhang. 2024. DistServe: disaggregating prefill and decoding for goodput-optimized large language model serving. In *Proceedings of the 18th USENIX Conference on Operating Systems Design and Implementation* (Santa Clara, CA, USA) *(OSDI'24)*. USENIX Association, USA, Article 11, 18 pages.