TokenPowerBench: Benchmarking the Power Consumption of LLM Inference

Chenxu Niu[1,2], Wei Zhang[3], Jie Li[2], Yongjian Zhao[2], Tongyang Wang[2], Xi Wang[4,5], Yong Chen[2]

1. NVIDIA Corporation, Santa Clara, CA, USA 2. Texas Tech University, Lubbock, TX, USA 3. Texas Advanced Computing Center, Austin, TX, USA
4. School of Integrated Circuits, Southeast University, Nanjing, China 5. National Center of Technology Innovation for EDA, Nanjing, China

AAAI-26 / IAAI-26 / EAAI-26
JANUARY 20–27, 2026 | SINGAPORE

BACKGROUND

Large language model (LLM) services now answer billions of queries per day, and industry reports show that inference, not training, accounts for more than 90% of total power consumption. However, existing benchmarks focus on either training/fine-tuning or performance of inference and provide little support for power consumption measurement and analysis of inference.

Energy-aware in LLM Inference

- LLMs are now central to large-scale production systems that translate, recommend, and converse across billions of user interactions.
- This rapid adoption has led universities, national laboratories, and cloud providers to set up dedicated GPU-backed inference services and AI testbeds.
- A fundamental systems question remains largely underexplored: ***"What is the power consumption of serving an LLM prompt?"***

Every Watt Counts as Cost

- According to a report on the operational lifecycle of LLMs from AWS, inference consumes more than 90% of the energy consumption.
- Furthermore, Gartner predicts that by 2028, over 80% of data center workload accelerators will be dedicated to inference, making a significant shift from the historically training-centric deployments.

LLM Inference Profiling & Power Analysis

- Recent studies begin to highlight the performance and energy cost of LLM inference.
- *MLPerf Power* treats LLM inference as a generic machine learning inference task and does not account for LLM-specific characteristics such as parallelism techniques and memory optimization.
- Other efforts propose power-aware profiling tools for general AI workloads, but they do not capture end-to-end inference flows in token-based generation models.
- In contrast, *TokenPowerBench* focuses specifically on ***LLM inference and provides phase-aware, token-level measurements*** aligned with model execution patterns. Our framework enables energy-normalized benchmarking (e.g., Joules per token) and supports reproducible multi-node configurations without requiring external metering hardware.
- It answers the question: ***"What is the real-world energy cost of running my massive, distributed model with my specific configuration on my available hardware?".***

Therefore, we present TokenPowerBench, the first lightweight and extensible benchmark specifically designed to quantify the power consumption and energy cost of LLM inference.

METHODOLOGY

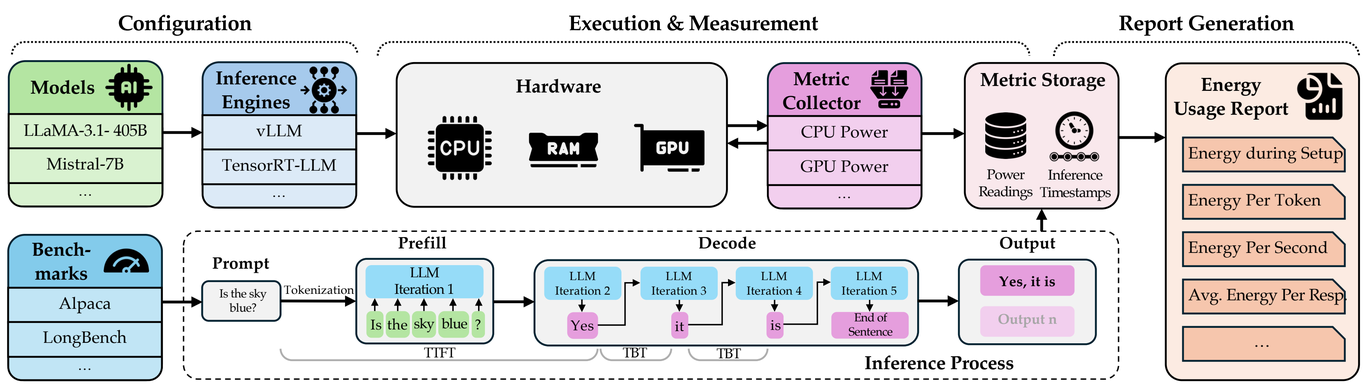

Figure 1: Overview of the TokenPowerBench Framework

Configuration

- The first task of *TokenPowerBench* is to help users set up the test environment by selecting the model to run, the inference engine to use, and prompts to feed into it.
- To keep this step lightweight, it exposes three plug-and-play modules: ***a model pool*** (choose any supported LLM), ***a prompt-dataset menu*** (select from Alpaca, LongBench, or custom prompts), and ***an inference-engine selector*** (vLLM, TensorRT-LLM, Transformers, or DeepSpeed).

Report Generation

- When an experiment ends, it collects the log data and turns the raw time-stamped samples into a compact summary.
- Users can get the data such as ***energy per token, energy per response, energy per second, peak power, and energy consumed during prefill stage***. Because every value is computed directly from the aligned samples, there is no need for post-hoc scaling or hand calculations.

Execution and Measurement

- **Spatial view (where the power is spent):**

During each run, *TokenPowerBench* samples power consumption from all major node components: ***GPU, CPU, DRAM***, and when sensors are available, the network interface and fan tray. GPU power is collected via ***NVML/DCGM***; CPU and DRAM power via ***Intel RAPL***; and full-node power via ***IPMI or a rack-mounted PDU***. Storing these readings side-by-side enables insights such as GPUs typically account for over 60% of total energy use, while fans contribute only a few percent.

- **Temporal view (when the power is spent):**

The same logger records two key stages of every inference: ***prefill***, when the model reads the input tokens, and ***decode***, when it produces new tokens. Each power sample is tagged with the stage that is active at that moment. After the run we integrate these tagged samples to obtain two clear numbers: energy consumed during prefill and energy consumed during decode. This separation reveals, for instance, that long prompts raise the prefill share, while large batch sizes increase the decode share.

EVALUATION

Experimental Setup

Hardware: a **NVIDIA H100 GPU cluster from NSF REPACSS Projects** (each node is equipment with 4 NVIDIA H100 GPUs with 94 GB memory, and Gold 6426Y CPU.
LLM families: **Llama 3, Falcon, Qwen**, and **Mistral.**
Prompt Datasets: **Alpaca** and **LongBench**

Main Results

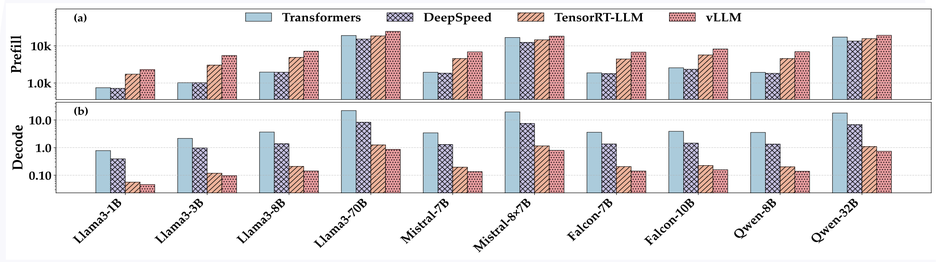

Figure 2: Prefill Energy and Decode Energy per Token Across Models and Inference Engines

- We tested all popular LLM models. Figure 2 compares energy consumed by prefill stage and total energy per token for **10 open-source models** (dense and MoE) served by four mainstream engines.
- Within each LLM family, energy rises faster than the parameter count. For LLaMA-3, moving from **1 B to 70 B** parameters increases energy per token by **7.3×**, even though parameter count grows **70×**.

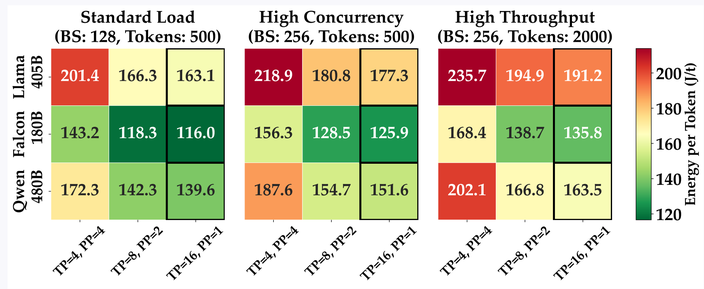

Figure 3: Heatmap of Energy per Token of SOTA models

- The heatmap figure presents a heatmap for three SOTA models: Llama 3 405B, Falcon180B, and Qwen 480B on 16 H100 GPUs under three workload profiles. **Across every workload the pure tensor parallelism setting delivers the best energy efficiency because long pipelines leave some GPUs idle.** The results show that parallelism tuning matters most in heavy, batch-oriented jobs.

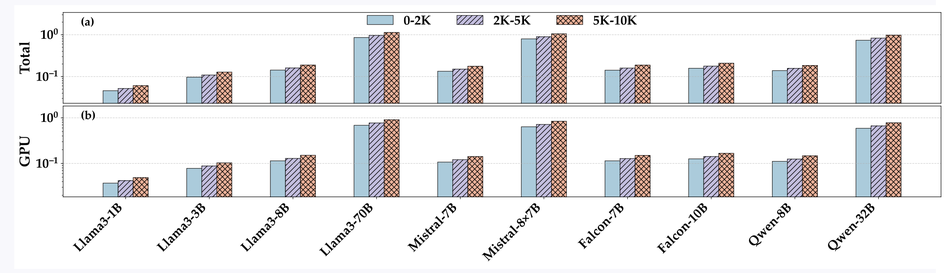

Figure 4: Comparison of Total and GPU Energy per Token at Varying Context Lengths

- For the largest dense model we tested (Llama3 70B), the jump from 2K to 10 K tokens raises energy per token by roughly a factor of three; medium models (e.g., Llama 3 8B, Mistral 7B) see a smaller but still clear rise.
- Longer prompts therefore hurt energy efficiency in two ways: they increase the joules spent before **the first output token appears,** and they **lower overall throughput** because GPUs stay busy on attention over a larger context window.

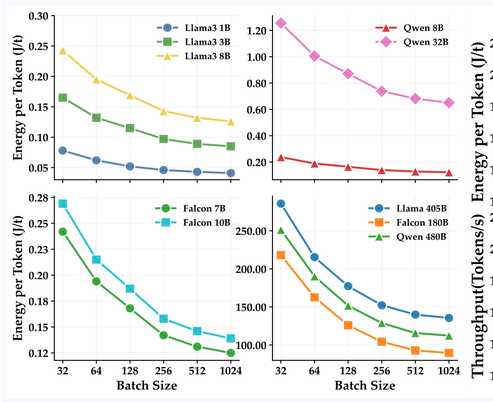

Figure 5: Comparison of Total Energy per Token Across Models

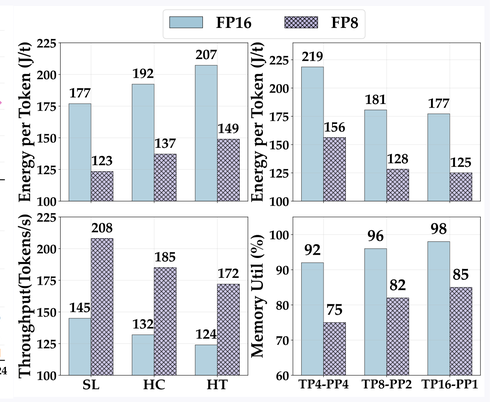

Figure 6: Comparison of Total Energy per Token Across Models

- We also plot the energy per token metrics for batch sizes from 32 to 1024 on six model families in figure 5. For every family, enlarging the batch lowers energy per token at first, because **the fixed set-up and kernel-launch overheads are spread over more tokens.**
- Figure 6 compares full-precision FP16 (16-bit Floating Point) inference with FP8 (8-bit Floating Point) weight quantization for Llama 3 405B. Across the three workload profiles: Standard Load, High Concurrency, and High Throughput, **quantization cuts energy per token by roughly 30%.**
- The saving is consistent regardless of the traffic pattern because the reduced-precision weights lower both memory traffic and arithmetic cost in every phase of generation. Measured per batch, total energy falls from about 45 kJ to 32 kJ under the heaviest load, mirroring the token-level reduction.

CONCLUSION

- **TokenPowerBench:** the first lightweight and extensible benchmark designed for LLM inference power consumption studies.
- **Key Innovation:** a **declarative configuration interface** covering model choice, prompt set, and inference engine; **a measurement layer** that captures GPU-, node-, and system-level power without specialized power meters; and **a phase-aligned metrics pipeline** that attributes energy to the prefill and decode stages.
- **Future work**: we will extend coverage beyond our current **H100 testbed** to other GPU architectures, including **the next NVIDIA generations and AMD accelerators**.

References

REPACSS | GitHub Repo | Video Presentation