Same Size, Different Costs: Phase-Level Energy Variations in Transformer Models during Code Generation

Lola Solovyeva[1]

[1]*University of Twente, Enschede, the Netherlands*

Abstract

AI-assisted tools are increasingly integrated into software development, augmenting workflows in code generation, bug fixing, testing, and documentation. However, their inference introduces extra energy costs that affect the sustainability of the software lifecycle. In this study, we measure phase-level energy consumption of LLMs, focusing on four transformer models of comparable size using HumanEval dataset for code generation under different batch sizes. Our findings show that models with similar parameter counts exhibit distinct energy consumption patterns across prefill and decoding phases. These results highlight that LLMs of the same architecture type and with similar parameter counts can still differ due to low-level implementation details, which should be considered when developing strategies to reduce energy consumption in software development.

1. Introduction

AI-assisted tools are increasingly integrated into software development processes [1]. In the context of software maintenance and evolution, these tools augment developer workflows in scenarios such as code generation, refactoring, bug detection, and testing [2]. While LLMs can accelerate those tasks, their inference processes introduce a non-trivial energy cost, particularly when used repeatedly in CI/CD pipelines or large-scale maintenance workflows. Research [3] suggested that OpenAI required 3,617 of NVIDIA's HGX A100 servers, with a tottal of 28,936 GPUs , to support ChatGPT, implying that it requires 564 MWh per day for its inference. Meanwhile, an estimate of 1,287 MWh was used in GPT-3 training phase. As a result, the overall sustainability of the software lifecycle now also depends on the efficiency of the AI tools that support them. While existing studies on LLM efficiency focus on architectural techniques, these approaches often treat inference as a uniform process [4]. In practice, inference consists of two distinct phases: **prefill**, that processes the input prompt and generates internal key/value representations (compute-bound), and **decoding**, that generates output tokens autoregressively using these cached representations (memory-bound).

In this work, we demonstrate that transformer models of similar sizes exhibit distinct energy consumption patterns across both phases. Hence, reducing the overall energy consumption of their inference requires model-specific optimization strategies.

2. Methodology

To record energy measurements per phase, we adopted the method originally proposed by Babakol et al. [5]. The method involves two parallel processes: (1) collecting GPU energy samples with pyNVML every 0.01 seconds along with their timestamps, and (2) recording timestamps at the start and end of generating each token. There was no other process running on the same GPU. The timestamps are then aligned to measure the energy consumption of each phase.

Four widely used transformer models with roughly similar parameter counts were selected from Hugging Face, ensuring they could be accommodated on an NVIDIA A10 GPU (24 GB RAM): Llama3.2

Woodstock'22: Symposium on the irreproducible science, June 07–11, 2022, Woodstock, NY

[*] You can use this document as the template for preparing your publication. We recommend using the latest version of the ceurart style.

o.solovyeva@utwente.nl (L. Solovyeva)
0009-0008-6903-7086 (L. Solovyeva)

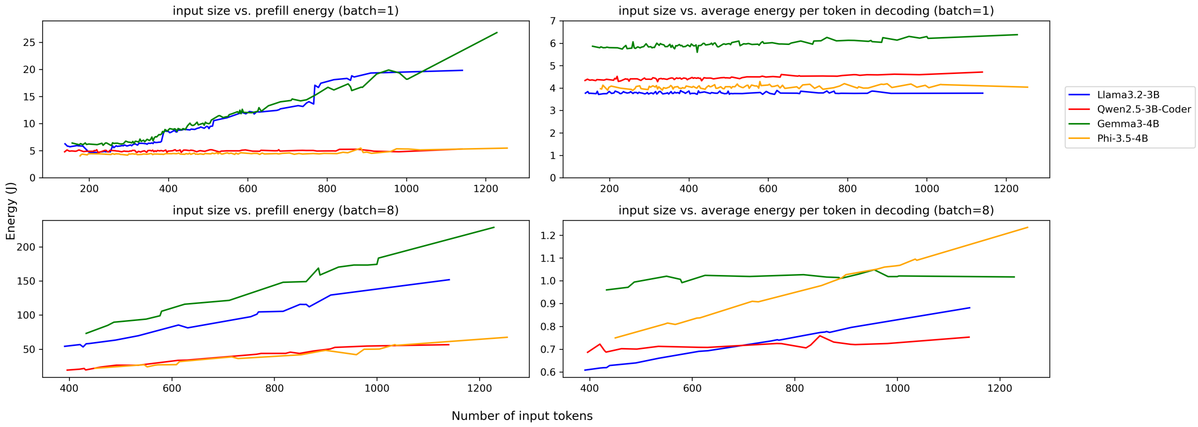

Figure 1: Influence of input tokens and batching on prefill phase costs and per-token costs in the decoding phase.

(3B), Qwen2.5-Coder (3B), Gemma3 (4B), and Phi3.5 (4B). We used the HumanEval dataset for code generation and evaluated the models with batch sizes of 1 and 8 to examine how a single request and batching, that increases the workload of the model, influence both phases.

3. Findings & Implications

Figure 1 shows the relationship between input prompt size and its impact on the prefill phase as well as per-token energy during the decoding stage for the models in this study. Overall, larger prompts and increased batch sizes lead to higher prefill costs. However, the magnitude of this increase varies across models, with some showing greater sensitivity to input size. Llama3.2 (3B) and Gemma3 (4B) exhibit a steeper increase compared to Qwen2.5 (3B) and Phi3.5 (4B), despite having similar parameter counts.

In regard to the influence of input size on the decoding stage, we can observe expected differences in costs between the models for a single request, since larger models would exhibit higher costs. A more interesting pattern emerges when processing batched requests, which increases the workload by combining multiple requests in one. The models respond differently: Phi3.5 (4B) and Llama3.2 (3B) show approximately a 1.5×increase in energy per token when the input grows from 400 to 1200 tokens, whereas the other two models are either unaffected or exhibit a much smaller increase.

These findings suggest that even among models of the same architecture type with similar parameter counts, their energy patterns differ across phases, indicating that these differences likely stem from low-level implementation details such as memory management and runtime optimizations. Furthermore, the choice of model within the software development lifecycle should depend on the specific task. For example, models that are less sensitive to input size may be better suited for tasks involving larger inputs, such as code translation, test or docstring generations.

References

[1] C. Ebert, P. Louridas, Generative ai for software practitioners, IEEE Software 40 (2023) 30–38. doi:10.1109/MS.2023.3265877.

[2] N. Alizadeh, B. Belchev, N. Saurabh, P. Kelbert, F. Castor, Language models in software development tasks: An experimental analysis of energy and accuracy, 2025. URL: https://arxiv.org/abs/2412.00329.

[3] A. de Vries, The growing energy footprint of artificial intelligence, Joule 7 (2023) 2191–2194. doi:https://doi.org/10.1016/j.joule.2023.09.004.

[4] M. F. Argerich, M. Patiño-Martínez, Measuring and improving the energy efficiency of large language models inference, IEEE Access 12 (2024) 80194–80207. doi:10.1109/ACCESS.2024.3409745.

[5] T. Babakol, Y. D. Liu, Tensor-aware energy accounting, in: Proceedings of the IEEE/ACM 46th International Conference on Software Engineering, ICSE '24, Association for Computing Machinery, New York, NY, USA, 2024. URL: https://doi.org/10.1145/3597503.3639156.