

Analysis of the impact of natural language selection in large language models on energy consumption

Tomasz Szymczyk^{1*} , Marcin Badurowicz¹ , Janusz Kozak¹ 

¹ Faculty of Electrical Engineering and Computer Science, Lublin University of Technology, ul. Nadbystrzycka 38A, 20-618 Lublin, Poland

* Corresponding author's email: t.szymczyk@pollub.pl

ABSTRACT

Large language model (LLM) inference is computationally intensive, yet the extent to which the interaction language affects energy consumption remains insufficiently understood. This study investigates whether inference energy consumption varies across six European languages: Polish, English, German, French, Spanish, and Russian. Three open-weight models, Llama3.1-8b, Gemma-3-12b, and Qwen2.5-14b, were evaluated using ten semantically equivalent prompts, each translated into the six languages and repeated three times. Experiments were conducted on a workstation-class server equipped with an AMD Radeon RX 7900 XT GPU and the Ollama inference framework. Whole-system power consumption was measured using a calibrated HIOKI PW3337 power meter, and energy efficiency was expressed as energy per generated token (EPT). Mean EPT varied substantially among models: Llama3.1-8b consumed 6.43 J/token, Qwen2.5-14b 10.23 J/token, and Gemma-3-12b 10.87 J/token. Kruskal–Wallis tests identified statistically significant between-language differences only for Llama3.1-8b; no significant differences were found for Qwen2.5-14b or Gemma-3-12b. However, regression analysis for Llama3.1-8b showed that response length was the dominant predictor of EPT, as fixed inference overhead was amortized over longer outputs. The findings indicate that apparent language-dependent differences in inference energy are driven primarily by generated-output length rather than by language itself. Energy efficiency also depends strongly on model architecture, tokenizer behavior, and inference implementation, highlighting the importance of response-length control and whole-system measurement in Green AI benchmarking.

Keywords: energy consumption, large language models, natural language processing, LLM, tokens.

INTRODUCTION

In recent years, large language models (LLMs) have become increasingly important in natural language processing (NLP). They offer powerful capabilities for text generation, translation, and language understanding, with applications ranging from everyday communication support to advanced data analysis. However, the growing capabilities of these models are accompanied by increasing demand for computational resources and, consequently, electricity. This creates new challenges for researchers and engineers seeking to improve energy efficiency.

To date, studies of the environmental impact of LLMs have focused primarily on model

architecture, parameter count, and training methods. However, relatively little attention has been paid to the effect of the natural language used to interact with a model. Differences in grammatical structure, word length, information redundancy, and other linguistic properties may translate into different computational loads.

This paper analyzes the impact of natural language choice on the energy consumption of LLMs during text generation. The study considered several languages with different linguistic properties and measured energy consumption during inference while keeping technical parameters constant. The findings may inform further work on improving the energy efficiency of AI-based systems and designing more

sustainable NLP services within the broader Green AI framework.

RELATED WORK

Natural languages differ in structural properties that can affect computational processing. Language complexity has been studied using various criteria and for different languages, including English [1], French [2], and Spanish [3]. Cross-linguistic complexity can be analyzed at several levels, including phonological, morphological, and syntactic complexity [4]. Table 1 summarizes processing characteristics, challenges, and tools relevant to the languages selected for this study.

The linguistic form of a prompt can affect how it is represented by a tokenizer. For semantically equivalent prompts in different languages, token counts can differ because tokenizer vocabularies and segmentation behavior vary across languages. With a standard BPE (byte pair encoding) tokenizer, agglutinative or polysynthetic languages (e.g., Turkish and Hungarian) may be tokenized into two or three times as many tokens as English [5]. Script is another relevant factor: although BPE tokenizers can operate across scripts, non-Latin scripts (e.g., Devanagari, Arabic, and Cyrillic) may yield up to four times as many tokens as Latin-script English for the same lexical content [6]. Biases in other popular tokenizers (GPT-cl100k, Gemini-o200k, SentencePiece) have also been analyzed, showing token-count inflation for low-resource languages [5]. Similar biases have been reported for OpenAI’s GPT-4, Google’s Gemini, Llama-2-13B, and Mistral-7B. In addition, language-specific grammatical structure

and typical verbosity may affect output length, potentially increasing total energy consumption. LLM performance also remains uneven across languages, with lower scores reported for many non-English languages [7].

Another important issue is energy consumption, particularly in the context of the Green AI movement. Large language models are used in many contexts and for various applications [8], but they are extremely compute-intensive during inference, leading to substantial energy consumption. Each query requires many floating-point operations; therefore, longer inputs or outputs generally require more energy [9], with energy requirements increasing linearly [10]. This suggests that, all else being equal, a language that results in more tokens may require more energy.

The choice between locally deployed LLMs and models available only through cloud providers (e.g., ChatGPT) depends on the intended use case. Edge computing brings computational resources closer to the point of execution [11] rather than relying exclusively on cloud providers. For confidential corporate data or trade secrets, locally deployed open-weight LLMs may be preferred [12]. However, such systems may not achieve the same levels of accuracy and can be affected by quantization [13], which is used to reduce hardware requirements and enable deployment on highly resource-constrained systems [14]. For cloud-based LLMs, users cannot directly measure energy consumption; typically, only token counts are observable [15], making actual energy use difficult to track.

For direct energy measurement, the MELODI framework [16] has been proposed. It uses the NVidia Smi and Scaphandre tools to estimate

Table 1. Main challenges and tools for processing the selected languages

Language	Tokenization and morphological profile	Main challenges	Tools required
Polish	Whitespace-delimited tokenization; morphologically rich, highly inflected language	Declension and inflectional endings	Lemmatization, morphological analysis
English	Predominantly whitespace-delimited tokenization; relatively limited inflection	Contractions and homonyms	Whitespace- and punctuation-based tokenization
German	Whitespace-delimited tokenization with productive compounding and inflection	Compound words and inflection	Compound-word decomposition
French	Whitespace-delimited tokenization with frequent elisions and apostrophes; inflectional variation	Elisions, apostrophes, and verb inflection	Handling elisions and apostrophes
Spanish	Whitespace-delimited tokenization with rich verb inflection and clitic pronouns	Verb inflection and attached pronouns	Tokenization with clitic-pronoun handling
Russian	Whitespace-delimited tokenization; morphologically rich, highly inflected language; Cyrillic script	Rich inflection, grammatical cases, verb aspect, and Cyrillic script	Lemmatization, morphological analysis

energy consumption from metrics reported by server hardware. Energy per token (EPT) has also been proposed [17] as a complement to traditional benchmarks. Energy measurements can be converted into estimates of carbon emissions [18], and methods for tracking greenhouse-gas emissions have also been discussed [19,20]. Carbon emissions are another key consideration [21], although artificial intelligence may also contribute to climate-change mitigation [22]. The EnergyMeter module, which combines several hardware metrics [19], has also been used.

Study [23] compared the energy consumption of selected large language models using server-class hardware equipped with NVIDIA H200 GPUs and NVML for reporting energy metrics directly from GPU hardware. This approach is recommended by the AI Energy Score initiative for standardization [24,25], but the use of server-class hardware may limit the applicability of such results to early-stage deployments in small and medium-sized enterprises. Cross-language comparisons have also been reported [26], indicating that low-resource languages can consume substantially more energy per request than high-resource languages such as English, with per-token energy varying by as much as 8.3-fold.

Finally, researchers have also explored ways to reduce the energy impact of LLM use [27,28], including avoiding wasted computations in automated systems such as GitHub Copilot, optimizing model and inference execution [29], improving data-center efficiency [30], and developing adaptive models for specific application domains [31].

The studies discussed above focus primarily on inference, that is, the generation of outputs by LLMs and SLMs. Model selection can itself affect energy use [32], but training is another major component of environmental impact and contributes to both energy use and water consumption [33]. Green AI initiatives [34] seek to reduce the carbon, energy, and water footprints of AI systems, and this study contributes to that broader research direction.

Limitations of existing studies include reliance on hardware-reported metrics, omission of whole-system energy consumption, and the use of server-class GPUs (e.g., Tesla V100), even though companies may initially deploy AI on consumer-class hardware. In this study, we compare inference energy consumption using a calibrated laboratory power meter, focusing on token generation for a small subset of high-resource

European languages. We also use an AMD GPU platform with the ROCm software stack as an alternative to the more common NVIDIA stack.

MATERIALS AND METHODS

The aim of this study was to investigate the factors influencing the energy consumption of large language model inference, with particular emphasis on the relationships among input language, tokenization efficiency, model architecture, and generated-output length. Previous studies have indicated that language models may exhibit different computational costs depending on linguistic characteristics; however, it remains unclear whether the observed differences stem from language itself or from indirect effects such as tokenizer behavior and variation in generated-output length. Therefore, this study evaluates energy consumption using the Energy per Token metric and analyzes whether language remains a statistically significant factor after controlling for response length.

Because changes in power consumption can be small, a high-precision laboratory power meter was used. Based on technical considerations and a review of measurement approaches reported in the literature, the following were developed:

- a test stand and measurement procedure,
- test scenarios.

Power measurements were performed using a HIOKI PW3337 power meter. The PW3337 is a three-channel AC/DC power meter designed for single-phase to 3P4W equipment, including motors, inverters, power conditioners, and power supplies. It is a high-precision instrument capable of measuring single- to three-phase lines. The meter covers frequencies from DC to inverter frequencies and supports voltages up to 1000 V and direct-input currents up to 65 A. According to the manufacturer's specifications, the instrument provides a fundamental accuracy of $\pm 0.15\%$ rdg. (at less than 50% of the range, $\pm 0.1\%$ rdg. $\pm 0.05\%$ f.s.) over a wide frequency band from DC and 0.1 Hz to 100 kHz. Accuracy is guaranteed for direct-input currents up to 65 A.

To prevent electric shock or short circuits at the input terminals, insulated solderless terminals were used. To prevent damage to the instrument or electric shock, dedicated $M6 \times 12$ mm screws were used to secure the voltage and current input

terminals, and $M3 \times 6$ mm screws were used to secure the safety cover supplied with the product, as shown in Figure 1. The complete connection diagram is shown in Figure 2 [35].

In the configuration used in this study, the measured voltage and current remained within the instrument's measurement range. Therefore, supplementary voltage or current transformers were not required, and the instrument was connected directly to the measurement lines.

The voltage input terminals were connected on the load side. The power measurement therefore includes losses due to the input resistance of the voltage terminals, which are less than 30 mW in this configuration.

The measurement frequency was set to 1 Hz. The primary display was configured to show the measured RMS voltage, the secondary display to show the RMS current, and the third display to show active power.

The computer-side system architecture is shown in Figure 3. The LLMs were installed on an isolated, purpose-built server with a 12-core AMD Ryzen 9 7900X processor, 64 GB of DDR5 RAM, and an AMD Radeon RX 7900 XT graphics card with 20 GB of VRAM. Each tested model fit entirely within GPU memory. The test server was selected as a workstation-class system representative of an LLM deployment in a small- or medium-sized enterprise.

Ollama server version 0.6 was used for inference. The Ollama inference software ran on Ubuntu 24.04 with ROCm version 6.4.1. All background services were disabled on the test machine, which was connected to a separate LAN accessible only to the control device.

Power-meter data acquisition and requests to the Ollama system were handled by scripts

developed by the authors and executed on a separate control device outside the measured circuit. The control device had two LAN interfaces: one connected to the power meter for data acquisition and the other to the inference server for sending requests and receiving responses, with the two connections operating on separate networks. To eliminate potential heat-induced GPU power throttling, all experiments were conducted in a climate-controlled environment.

To exclude the energy associated with loading a model into VRAM, energy consumption was measured only after the model had been fully loaded into GPU memory. The experimental procedure included a warm-up stage to load the model, followed by a series of prompts. Additionally, the model KV cache and conversational context were reset between successive measurements so that each query began with a fresh context while the model remained loaded. The keep-alive setting, which controls how long a model remains in memory, was set to 60 minutes. The eviction timer was refreshed each time a request was processed, ensuring that no tested model was evicted during the experimental runs.

The control device started recording immediately before a request was sent and stopped after the complete response and associated metadata had been returned; streaming responses were not used. The server's idle (baseline) power was included in the measured power consumption.

No maximum output-token limit was imposed on the prompts; unconstrained generation was used to better represent realistic LLM use.

Three large language models were evaluated:

- Llama3.1-8b is a dense, decoder-only Transformer language model developed by Meta with approximately 8 billion parameters. It



Figure 1. Connection of the HIOKI PW3337 to the measurement line

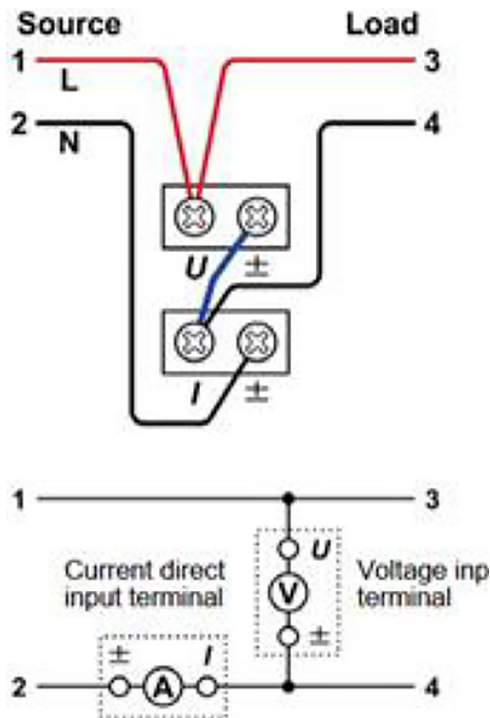


Figure 2. HIOKI PW3337 connection diagram [35]

is designed as a general-purpose multilingual model [36]. Although released in 2024, it remains in widespread use. The variant used in this study was obtained from Ollama's model directory with Q4_K_M quantization; its model ID was 46e0c10c039e.

- Gemma-3-12b is a 12-billion-parameter model in Google's Gemma 3 family of light-weight open models. The model supports a context length of at least 128k tokens and uses an attention design with a higher proportion of local attention layers to reduce the memory requirements of long-context inference. Gemma 3 models were trained using

distillation and extensive post-training and are known for strong multilingual capabilities [37]. The variant used in this study also used Q4_K_M quantization; its model ID in Ollama's directory was f4031aab637d.

- Qwen2.5-14b is a dense 14-billion-parameter general-purpose language model from Alibaba's Qwen2.5 family. It is designed to support a broad range of tasks and achieves strong results on many benchmarks [38]. Q4_K_M quantization was also used for this model; its model ID was 7cdf5a0187d5.

The models were selected to represent a popular model (Llama3.1-8b), a model with strong multilingual capabilities (Gemma-3-12b), and a model with strong benchmark performance (Qwen2.5-14b). At the time the experiments were conducted, these were among the newest available model versions; given the rapid pace of development in the field, they are likely to be superseded by newer models.

All models were also selected for compatibility with the software platform and their ability to fit within the available VRAM. We used Ollama's Q4_K_M quantized variants, downloaded using commands such as "ollama pull qwen2.5:14b" (and analogous commands for the other models), reflecting a typical user workflow.

The tested models differed in parameter count, allowing us to assess whether model size alone, without accounting for architecture and design, is linearly associated with energy consumption.

Although all tested models belong to the same broad category of Transformer-based LLMs, implementation differences may affect energy efficiency during inference. Parameter count, architectural modifications, tokenizer design,

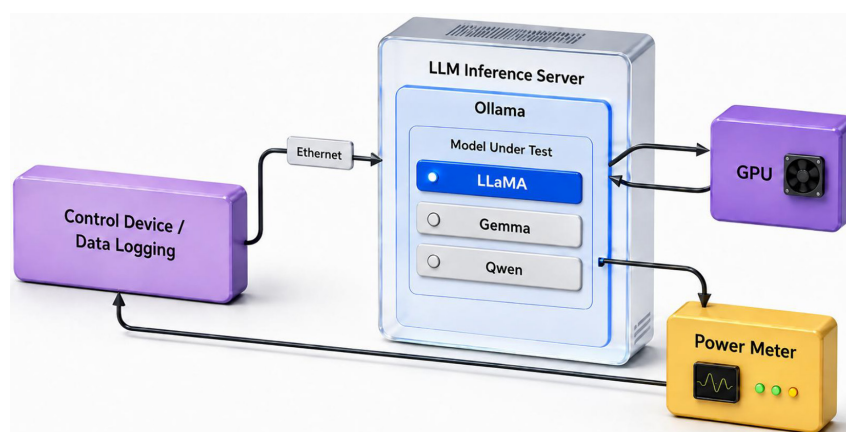


Figure 3. Experimental system architecture

attention mechanisms, quantization method, and inference-engine optimization can all influence GPU utilization and execution time. Therefore, differences observed among models should not be interpreted solely as consequences of parameter count, but rather as the combined effects of model architecture and software implementation.

The experiment used a set of 10 questions, shown in Table 2. Each question was translated into Polish, English, German, French, Spanish,

and Russian. Table 3 presents the token counts for each question and language after tokenization.

Because the analyzed models use different tokenizers, prompt token counts varied among models. Table 3 shows that English had the lowest mean prompt token count for all three models (11.2, 11.0, and 11.2, respectively), whereas Polish had the highest mean in all three cases. For Russian, which uses Cyrillic script, prompt token counts were not consistently higher than

Table 2. Questions used in the study

Question number	Prompt content	Field of knowledge (general)
P1	Explain the principles of quantum mechanics in simple words.	Physics, Mathematics
P2	Describe the history of the Internet and its impact on society.	History, Sociology
P3	How does artificial intelligence work and where is it used?	Computer Science, Sociology
P4	What is cryptography and why is it important for data security?	Computer science, Cryptography,
P5	Explain the differences between biological and artificial neurons.	Biology, Computer science
P6	How does blockchain work and what are its applications?	Mathematics, Computer science
P7	Describe the history of spaceflight and its future.	Astronomy, History
P8	What is augmented reality and what are its applications in education?	Computer science, Education
P9	What are the main challenges posed by climate change?	Environmental protection, History, Sociology
P10	Explain how renewable energy sources such as photovoltaics work.	Electrical engineering, Sociology

Table 3. Token counts for all questions

Llama3.1-8b										
Language	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Polish	19	22	19	23	23	12	17	24	19	25
German	21	19	18	18	20	12	15	19	19	21
French	18	18	18	20	20	12	17	20	17	22
Spanish	19	13	14	20	17	13	14	20	19	18
English	11	12	11	12	10	10	10	12	10	14
Russian	16	19	14	15	18	10	17	18	16	23
Gemma3-12b										
Polish	19	17	13	17	18	9	15	18	16	21
German	17	15	14	15	16	11	12	15	16	17
French	14	16	13	21	14	10	14	20	11	19
Spanish	13	12	11	18	14	10	13	15	12	17
English	10	12	11	12	9	10	10	12	10	14
Russian	14	13	11	14	16	10	12	15	11	21
Qwen2.5-14b										
Polish	19	20	17	17	21	10	16	24	16	25
German	21	19	18	19	20	12	15	19	19	21
French	18	18	18	20	20	12	17	19	17	22
Spanish	19	13	14	19	17	13	14	19	19	18
English	11	12	11	12	10	10	10	12	10	14
Russian	18	19	13	17	21	12	19	18	16	23

those of the Latin-script languages; however, script type was not modeled independently as a predictor. The prompts were intentionally designed to be short and semantically neutral. The objective was to minimize the influence of the prompt-processing phase and focus the analysis on the autoregressive generation phase, during which most computational operations occur. Because the generated responses were substantially longer than the input prompts, the energy required for prompt processing was expected to make only a minor contribution to total inference energy.

As shown in Figure 4, the prompts differed not only in token count but also in word and character counts. Differences in word count were relatively small, whereas differences in character count were more pronounced.

RESULTS AND DISCUSSION

The null hypothesis was that EPT would not differ among languages. Under this hypothesis, the energy required to generate a response should scale with the number of generated tokens rather than vary systematically by language. Although token counts may differ across languages, as shown in Table 3, overall EPT should remain comparable.

For each model, 180 responses were collected (10 questions $\times$ 6 languages $\times$ 3 repetitions), together with generation metadata, including the number of generated tokens, processing start and stop times, and total processing time.

Power values were recorded at each time-stamp. These measurements were used to calculate, for each model and language, the total number of generated tokens and the total energy consumed (J), from which the energy per token (EPT) metric [17] was obtained. In this study, EPT was calculated discretely over time, as shown in Equation 1:

$$E_t = \frac{\sum_{k=1}^n W_c \cdot t}{T_p} \quad (1)$$

where: E_t denotes energy per token; W_c is the power value read from the meter; T_p is the number of generated tokens; t is the sampling interval (1 s); and T_0 and T_n denote the beginning and end of the observation interval, respectively.

Thus, E_t is expressed in joules per token (J/token), because:

$$\left[\frac{J}{\text{token}} \right] = \frac{[W] \times [s]}{[\text{tokens}]} \quad (2)$$

On average, Gemma-3-12b consumed 10.87 J/token, Llama3.1-8b 6.43 J/token, and Qwen2.5-14b 10.23 J/token. The corresponding standard deviations were 0.24, 0.34, and 0.15, respectively. Results broken down by language are presented in Table 4. They show that response token counts vary across languages, consistent with the factors discussed earlier. They also show that, on average, Gemma-3-12b, despite having fewer parameters, consumes more Energy per Token than Qwen2.5-14b.

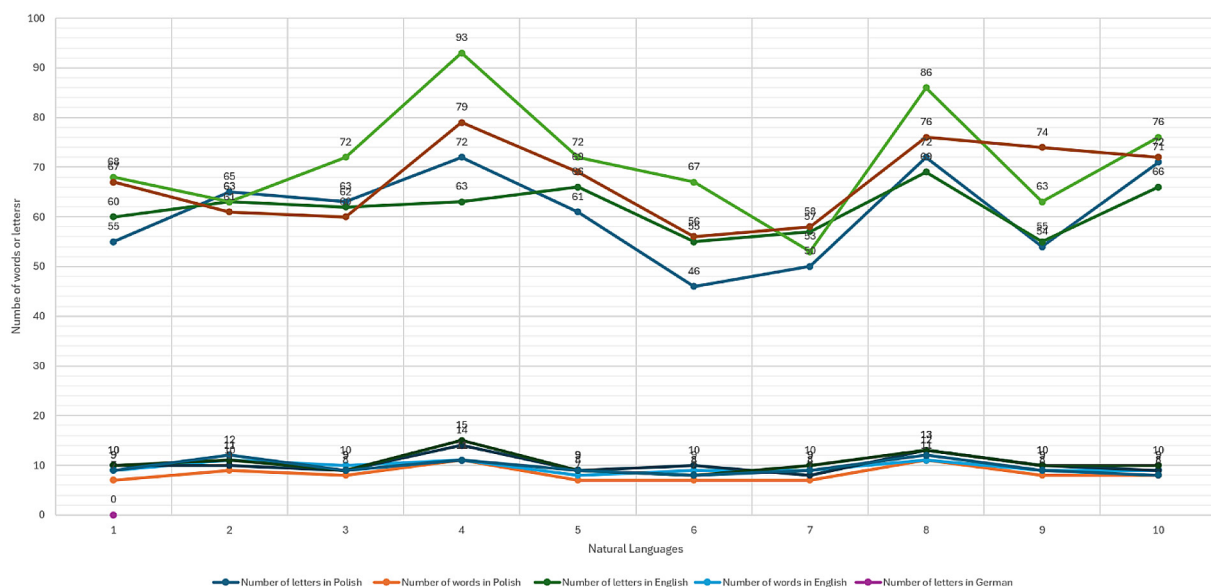


Figure 4. Prompt length characteristics across the selected languages

Table 4. Total generated tokens, total energy, average EPT, and average response length for all tested models

Language	Tokens generated in Total	Energy in Total [J] (1)	Energy Per Token [J/token]	Average number of tokens in response
Llama3.1-8b				
Polish	11352	74223.39	6.54	378.40
English	18504	112457.10	6.08	616.80
German	16099	97868.10	6.08	536.63
French	7260	49878.79	6.87	242.00
Spanish	11369	73516.29	6.47	378.97
Russian	11910	78178.57	6.56	397.00
Gemma3-12b				
Polish	36607	399023.81	10.90	1220.23
English	39665	433654.17	10.93	1322.17
German	38403	408472.17	10.64	1280.10
French	34329	372061.31	10.84	1144.30
Spanish	34171	385938.64	11.29	1139.03
Russian	35941	381075.21	10.60	1198.03
Qwen2.5-14b				
Polish	20231	212129.73	10.49	674.37
English	17923	181940.09	10.15	597.43
German	20602	208595.49	10.13	686.73
French	21282	216050.15	10.15	709.40
Spanish	19908	202483.99	10.17	663.60
Russian	18713	192626.98	10.29	623.77

For Qwen2.5-14b, the number of response tokens was higher than for English in every other language, by up to 18% for French. To visualize between-language differences, the analysis focused on Energy per Token, as shown in Figure 5. The observed differences were small in magnitude and were examined more rigorously using

per-question EPT values. Normality was assessed with the Shapiro–Wilk test.

For all three models, the EPT distributions deviated significantly from normality (Llama3.1-8b: $W = 0.7279$, $p < .001$; Qwen2.5-14b: $W = 0.9347$, $p < .001$; Gemma-3-12b: $W = 0.5882$, $p < .001$). Given these departures from normality,

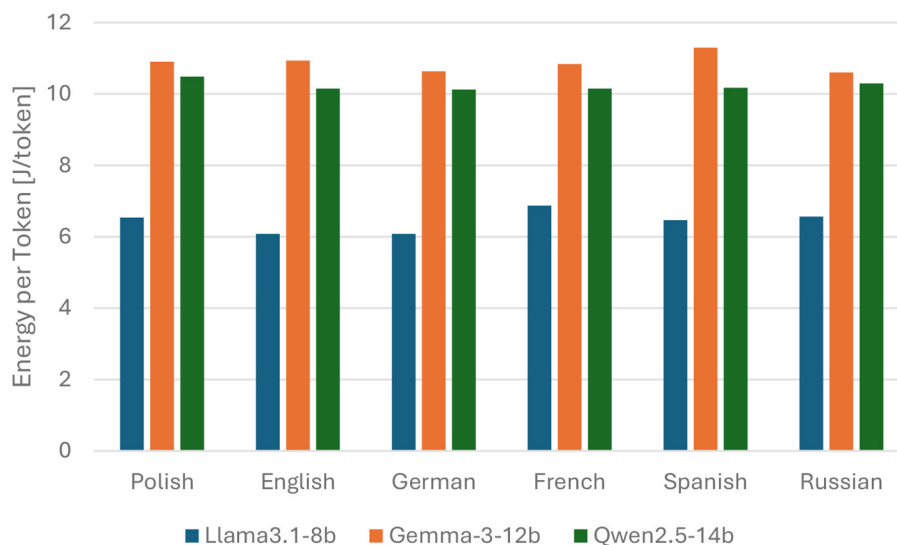


Figure 5. Average energy per token values across languages and models

all between-language comparisons were performed using the nonparametric Kruskal–Wallis rank-based test, followed by pairwise Dunn tests with Holm correction.

The results of the Kruskal–Wallis tests are summarized in Table 5. For Llama3.1-8b, language had a statistically significant effect on Energy per Token, $H(5) = 28.28$, $p < .001$. The associated effect size was $\varepsilon^2 = 0.134$, corresponding to a moderate-to-large effect according to conventional criteria (small ≈ 0.01 , medium ≈ 0.06 , large ≈ 0.14). This indicates that language accounted for a meaningful proportion of the variance in per-token energy consumption in the unadjusted comparison.

In contrast, no statistically significant between-language differences were detected for Qwen2.5-14b, $H(5) = 9.38$, $p = .095$, $\varepsilon^2 = 0.025$ (small effect), or for Gemma-3-12b, $H(5) = 4.59$, $p = .469$, $\varepsilon^2 = -0.0026$.

Thus, the association between language and per-token energy consumption was model-specific: a statistically significant language effect was observed for Llama3.1-8b, whereas Qwen2.5-14b and Gemma-3-12b showed no statistically detectable differences among languages.

Given the significant omnibus test, pairwise Dunn tests with Holm correction were conducted for Llama3.1-8b. Of the 15 possible pairwise comparisons, five remained statistically significant at the conventional threshold of $\alpha = 0.05$, as shown in Figure 6. All significant contrasts involved either English or German and one of the higher-EPT languages: English differed significantly from French ($p < .001$), Russian ($p = .007$), Polish ($p = .015$), and Spanish ($p = .023$), while German differed significantly from French ($p = .007$). No other comparisons were significant (all Holm-adjusted $p \geq .118$). In particular, English and German did not differ from each other ($p = 1.000$), and the remaining languages (French, Polish, Russian, and Spanish) did not differ significantly among themselves.

For Qwen2.5-14b, although the omnibus test was not significant, pairwise comparisons were also examined for completeness; none was statistically significant (minimum Holm-adjusted $p = .240$, German vs. Russian).

For Gemma-3-12b, all pairwise comparisons yielded a Holm-adjusted p -value of 1.000. These null post hoc results are consistent with the

Table 5. Summary of distributional tests and Kruskal–Wallis comparisons of energy per token across languages

Model	Shapiro–Wilk W	Shapiro–Wilk p	$H(df = 5)$	p	ε^2	Interpretation
Llama-3.1-8b	0.7279	< 0.001	28.28	< 0.001	0.134	Significant, moderate-to-large effect
Qwen2.5-14b	0.9347	< 0.001	9.38	.095	0.025	Not significant, small effect
Gemma-3-12b	0.5882	< 0.001	4.59	.469	-0.0026	Not significant, negligible effect

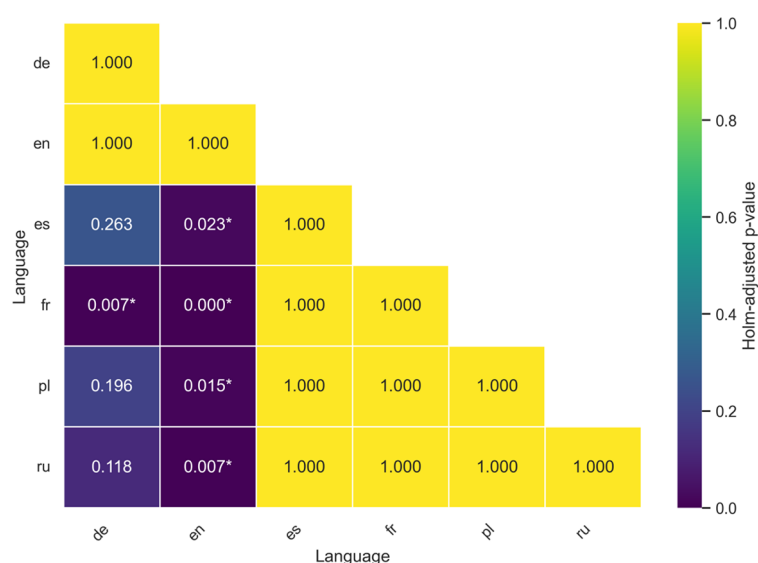


Figure 6. Results of pairwise Dunn tests with Holm correction for Llama3.1-8b; values marked with * indicate statistically significant differences

non-significant omnibus tests and provide no evidence of language-related differences in per-token energy consumption for either of these two models.

For Llama3.1-8b, Energy per Token was negatively associated with response length (English: longest responses, 616.8 tokens, lowest EPT; French: shortest responses, 242 tokens, highest EPT). This pattern suggested the presence of fixed per-response costs, such as computational overhead, that are amortized over longer outputs. To test this hypothesis, we fitted an ordinary least squares (OLS) regression. EPT (J/token) was used as the dependent variable, token count as a continuous covariate, and language as a categorical predictor ($n = 180$; German served as the reference language). The analysis assessed whether language remained associated with EPT after accounting for variation in output length.

The model was statistically significant, $F(6, 173) = 8.00$, $p < .001$, and explained $R^2 = 0.217$ (adjusted $R^2 = 0.190$) of the variance in EPT. Response length was the dominant predictor: each additional generated token was associated with an approximately 0.0038 J/token decrease in Energy per Token. After response length was accounted for, all between-language contrasts were statistically non-significant.

In this analysis, response token count emerged as the primary predictor, indicating that the previously observed differences were driven largely by variation in response length rather than by linguistic properties.

However, the results also show that LLM energy efficiency per token is not constant; it varies

across models and architectures and is not solely a function of parameter count. These findings are consistent with those reported in [30], where J/token values for high-resource languages were comparable. Our experiment did not include typical low-resource languages. In addition, our measurements concern whole-system energy consumption rather than GPU energy alone, and the observed model-level differences therefore reflect the combined behavior of the complete inference system.

The tokenizer is an important intermediate factor because different models convert the same linguistic content into different token sequences. More efficient tokenization can reduce the number of processing steps required for the same content; however, tokenization alone cannot fully explain the observed energy differences. The regression analysis indicates that generated-output length has a stronger influence than input tokenization efficiency.

The inference-software implementation can also influence energy consumption. In this study, the Ollama framework manages model execution, GPU memory allocation, and computational scheduling; therefore, differences in model support and optimization may affect hardware utilization. Measured energy values should thus be interpreted as the combined result of model architecture, tokenizer behavior, and inference implementation rather than as a direct measure of theoretical computational complexity. Disentangling these factors (tokenizer, architecture, and inference stack) was outside the scope of the present

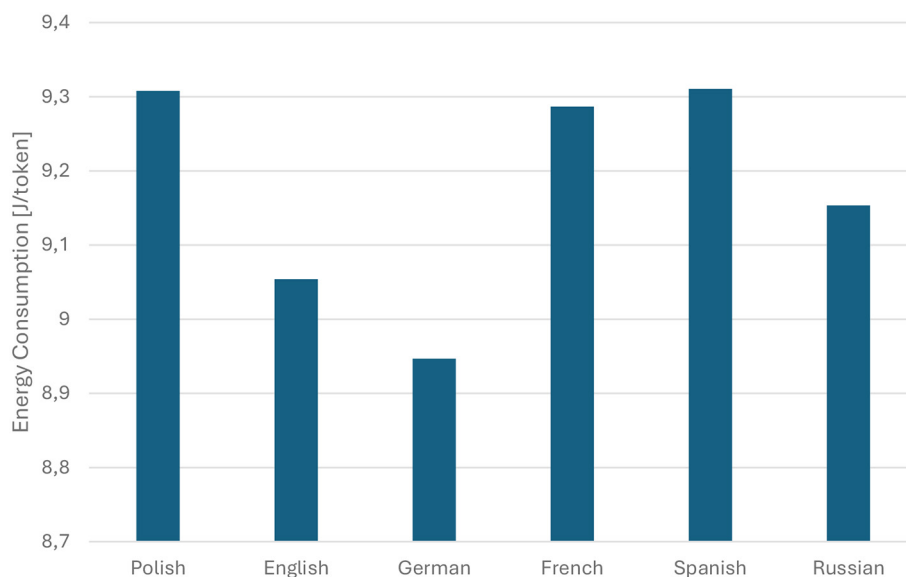


Figure 7. The average energy per token across all languages and models

experimental design and represents an important direction for future work.

In summary, across the three LLMs tested, Energy per Token was influenced primarily by output length rather than language. For Llama3.1-8b, EPT decreased as output length increased, consistent with fixed computational overhead being amortized over longer responses.

The results indicate that small language-dependent differences are visible in the raw energy measurements shown in Figure 7; however, after accounting for the number of generated tokens, these effects are substantially reduced.

CONCLUSIONS

This study experimentally demonstrates that observed language-dependent differences in LLM inference energy are largely attributable to tokenization and generated-output length. A second key finding is that energy efficiency per token varies with output length because inference includes a fixed computational overhead. Consequently, a large number of short queries may be less energy-efficient per token, whereas the fixed overhead is amortized over longer generations. Parameter count alone did not predict energy use: in the present experiments, Gemma-3-12b consumed more energy per token than the larger Qwen2.5-14b model.

To reduce the environmental impact of LLM deployment, priority should be given to optimizing model architecture, inference configuration, and response-length management.

REFERENCES

1. Kwapien J, Drozd S, Orczyk A. Linguistic complexity: English vs. Polish, Text vs. Corpus. *Acta Phys Pol A* 2010;117:716–20. <https://doi.org/10.12693/APhysPolA.117.716>
2. Pimpied V, Beauchemin D, Khoury R. Quantifying French Document Complexity. *Proceedings of the Canadian Conference on Artificial Intelligence* 2022. <https://doi.org/10.21428/594757db.8b7cd94b>
3. Gutierrez-Vasques X, Mijangos V. Comparing morphological complexity of Spanish, Otomi and Nahuatl. In: Becerra-Bonache L, Jiménez-López MD, Martín-Vide C, et al., editors. *Proceedings of the Workshop on Linguistic Complexity and Natural Language Processing*, Santa Fe, New-Mexico: Association for Computational Linguistics; 2018, p. 30–7.
4. Miestamo M. Grammatical complexity in cross-linguistic perspective. In: Miestamo M, Sinnemäki K, Karlsson F, editors. *Language Complexity: Typology, contact, change*, John Benjamins Publishing Company; 2008, p. 23–41. <https://doi.org/10.1075/slcs.94.04mie>
5. Lundin JM, Zhang A, Karim N, et al. The Token Tax: Systematic Bias in Multilingual Tokenization. *Proceedings of the 7th Workshop on African Natural Language Processing (AfricaNLP 2026)*, Stroudsburg, PA, USA: Association for Computational Linguistics; 2026, p. 103–12. <https://doi.org/10.18653/v1/2026.africanlp-main.10>
6. Dixit A, Dixit S. The script tax: Measuring tokenization-driven efficiency and latency disparities in multilingual language models. *arXiv* 2026. <https://arxiv.org/abs/2602.11174>
7. Li Z, Shi Y, Liu Z, et al. Language Ranker: A Metric for Quantifying LLM Performance Across High and Low-Resource Languages. *Proceedings of the AAAI Conference on Artificial Intelligence* 2025;39:28186–94. <https://doi.org/10.1609/aaai.v39i27.35038>
8. Klekowicki M, Szymański GM, Waligórski M, et al. Application of large language models in diagnostics and maintenance of aircraft propulsion systems. *Advances in Science and Technology Research Journal* 2024;19:304–20. <https://doi.org/10.12913/22998624/196264>
9. Poddar S, Koley P, Misra J, et al. Towards Sustainable NLP: Insights from Benchmarking Inference Energy in Large Language Models. *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, vol. 1, Stroudsburg, PA, USA: Association for Computational Linguistics; 2025, p. 12688–704. <https://doi.org/10.18653/v1/2025.naacl-long.632>
10. Maliakel PJ, Ilager S, Brandic I. Characterizing LLM inference energy-performance tradeoffs across workloads and GPU scaling. *arXiv*, 2026. <http://arxiv.org/abs/2501.08219>
11. Palaniappan SC, Ponnuswamy PP. Task offloading in edge computing using integrated particle swarm optimization and genetic algorithm. *Advances in Science and Technology Research Journal* 2024;19:371–80. <https://doi.org/10.12913/22998624/195658>
12. Badurowicz M, Skulimowski SP, Laskowski M. Feasibility of using low-parameter local LLMs in answering questions from enterprise knowledge base. *Applied Computer Science* 2024;20:175–91. <https://doi.org/10.35784/acs-2024-46>
13. Tuggener L, Sager P, Taoudi-Benchekroun Y, et al. So you want your private LLM at home? A survey and benchmark of methods for efficient GPTs.

- 2024 11th IEEE Swiss Conference on Data Science (SDS), IEEE; 2024, p. 205–12. <https://doi.org/10.1109/SDS60720.2024.00036>
14. Seymour L, Kutukcu B, Baidya S. Large language models on small resource-constrained systems: performance characterization, analysis and trade-offs. *arXiv*, 2024. <http://arxiv.org/abs/2412.15352>
15. The tricky task of calculating AI's energy use n.d. <https://www.economist.com/science-and-technology/2025/04/09/the-tricky-task-of-calculating-ai-energy-use> (accessed July 3, 2026).
16. Husom EJ, Goknil A, Shar LK, et al. The price of prompting: Profiling energy use in large language models inference. *arXiv*, 2026. <http://arxiv.org/abs/2407.16893>
17. Wilhelm P, Wittkopp T, Kao O. Beyond Test-Time Compute Strategies: Advocating Energy-per-Token in LLM Inference. *Proceedings of the 5th Workshop on Machine Learning and Systems*, vol. 1, New York, NY, USA: ACM; 2025, p. 208–15. <https://doi.org/10.1145/3721146.3721953>
18. Gaddam P, Kulkarni P. Estimating the Carbon Footprint of Small Language Models Powering Edge Device Applications. *2024 Conference on AI, Science, Engineering, and Technology (AIXSET)*, IEEE; 2024, p. 185–8. <https://doi.org/10.1109/AIXSET62544.2024.00034>
19. Naidu R, Diddee H, Mulay A, et al. Towards quantifying the carbon emissions of differentially private machine learning 2021. *arXiv*, 2021. <http://arxiv.org/abs/2107.06946>
20. Luccioni AS, Viguier S, Ligozat A-L. Estimating the carbon footprint of BLOOM, a 176B parameter language model. *J Mach Learn Res* 2023;24. <https://doi.org/10.5555/3648699.3648952>
21. The Environmental Impacts of AI – Primer n.d. <https://huggingface.co/blog/sasha/ai-environment-primer> (accessed July 3, 2026).
22. Kaack LH, Donti PL, Strubell E, et al. Aligning artificial intelligence with climate change mitigation. *Nat Clim Chang* 2022;12:518–27. <https://doi.org/10.1038/s41558-022-01377-7>
23. de Zarzà I, Liz M, de Curtò J, et al. Energy-aware multilingual evaluation of large language models. *Electronics (Basel)* 2026;15:1395. <https://doi.org/10.3390/electronics15071395>
24. AI Energy Score | AI Energy Score: Initiative to establish comparable energy efficiency ratings for AI models. n.d. <https://huggingface.github.io/AIEnergyScore/> (accessed August 9, 2026).
25. Gupta U, Kim YG, Lee S, et al. Chasing carbon: The elusive environmental footprint of computing. *IEEE Micro* 2022;42:37–47. <https://doi.org/10.1109/MM.2022.3163226>
26. Deng N, Shen A, Feng Y, et al. The language-energy divide: measuring energy costs of multilingual LLM inference 2026. *arXiv*, 2026. <http://arxiv.org/abs/2606.21869>
27. Nik A, Riegler MA, Halvorsen P. Energy-conscious LLM decoding: Impact of text generation strategies on GPU energy consumption. *arXiv*, 2025. <http://arxiv.org/abs/2502.11723>
28. Coignon T, Quinton C, Rouvoy R. Green My LLM: Studying the key factors affecting the energy consumption of code assistants. *arXiv*, 2024. <http://arxiv.org/abs/2411.11892>
29. Hisaharo S, Nishimura Y, Takahashi A. Optimizing LLM inference clusters for enhanced performance and energy efficiency. *TechRxiv*, 2024. <https://doi.org/10.36227/techrxiv.172348951.12175366/v1>
30. Stojkovic J, Choukse E, Zhang C, et al. Towards greener LLMs: Bringing energy-efficiency to the forefront of LLM inference 2024. *arXiv*, 2024. <http://arxiv.org/abs/2403.20306>
31. Wiliński A, Datta A, Kubica B. Adaptive modeling algorithm for energy optimization in nanosensor networks for smart IoT systems. *Advances in Science and Technology Research Journal* 2025;19:292–306. <https://doi.org/10.12913/22998624/208842>
32. Klesel M, Messer U. Good for the Planet, Bad for Me? Intended and Unintended Consequences of AI Energy Consumption Disclosure. *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, vol. 1, New York, NY, USA: ACM; 2026, p. 1–14. <https://doi.org/10.1145/3772318.3791015>
33. Li P, Yang J, Islam MA, et al. Making AI Less “Thirsty.” *Commun ACM* 2025;68:54–61. <https://doi.org/10.1145/3724499>
34. Schwartz R, Dodge J, Smith NA, et al. Green AI. *Commun ACM* 2020;63:54–63. <https://doi.org/10.1145/3381831>
35. HIOKI PW3336 Measurement Guide. n.d. [Internet] <https://www.hioki.com/download/28782> (accessed July 3, 2026).
36. Grattafiori A, Dubey A, Jauhri A, et al. The Llama 3 Herd of Models 2024. *arXiv*, 2024. <http://arxiv.org/abs/2407.21783>
37. Gemma Team, Kamath A, Ferret J, et al. Gemma 3 technical report 2025. *arXiv*, 2025. <http://arxiv.org/abs/2503.19786>
38. Yang A, et al. Qwen2.5 technical report 2025. *arXiv*, 2025. <http://arxiv.org/abs/2412.15115>