LLMs on the Edge: Quality, Latency, and Energy Efficiency

Sebastian Bast[1], Lejla Begic Fazlic[1], Stefan Naumann[1], and Guido Dartmann[1]

Abstract: Generative Artificial Intelligence has become an integral part of many people's lives. Large Language Models (LLMs) are gaining increasing popularity in science and society. While it is well known that training these models requires significant energy, inference also contributes to their total energy demand. Therefore, we analyze how to use them as sustainably as possible by investigating the efficiency of inference, especially on local hardware with limited computing power. We develop metrics for quantifying the efficiency of LLMs on the edge, focusing on the most influential factors *quality*, *time*, and *energy*. We compare the performance of three different state of the art generative models on the edge and assess the quality of the generated text, the time used for text creation and the energy demand down to the token level. The models achieve between $73,3\%$ and $85,9\%$ on the quality level, generate $1,83$ to $3,51$ tokens per second while consuming between $0,93$ and $1,76$ mWh of energy per token on a single-board computer without GPU-support. The findings of this study demonstrate that generative models can produce satisfactory outcomes on edge devices. However, a thorough efficiency evaluation is recommended before deploying them in production environments.

Keywords: Large Language Models, Generative Artificial Intelligence, Edge Devices, Efficiency

1 Introduction

While we ask ourselves if machines can ever be as smart as humans, a single-board computer smaller than the palm of our hands may already have generated a plausible-sounding answer. Until recently, resource limitations made it difficult for edge devices to effectively handle generative AI models. However, with the rise of Large Language Models (LLMs) entering the public domain, rapid advancements in generative AI research, and the momentum of the open-source movement, edge devices can now overcome these challenges. In this paper, we examine their performance by assessing the efficiency of three generative models operating on a resource-limited edge device measuring the quality of generated texts, runtime during text generation, and energy demand during inference. We calculate their efficiency using our metrics customized for LLM inference on the edge. While a model like Llama 3 [Me24] with 8 billion parameters was trained for 1,3 million GPU hours, we aim to determine how much time and energy this specific model requires to generate a single useful token on resource-constrained hardware. This article is structured as follows: We begin with a brief state-of-the-art overview, followed by our methodology. Next, we develop metrics and present experimental results. We summarize our findings in the discussion and conclude in the final chapter. The complete source code, data, and results are publicly available[2].

[1] University of Applied Sciences Trier, Environmental Campus Birkenfeld, Institute for Software Systems, PO-Box 1380, 55761 Birkenfeld, Germany, s.bast@umwelt-campus.de; l.begic@umwelt-campus.de; s.naumann@umwelt-campus.de; g.dartmann@umwelt-campus.de

[2] GitLab Repository, https://gitlab.rlp.net/ISS/llms-on-the-edge, accessed: 06/24/2024

2 State of the Art

Many Transformer-based models [Va17] are utilized in various language-based applications, such as conversational agents. LLMs like GPT-4 [Op24] demonstrate human-level performance on numerous professional and academic benchmarks. However, their billions of parameters typically render them unsuitable for resource-limited devices. Recently, techniques like quantization [Li24] and pruning [Su24] have significantly optimized model efficiency by reducing computational requirements and the number of parameters. Consequently, small LLMs like Llama 3 [Me23], Mistral [Ji23], and Phi-3 [Ab24] now range from 3,8 to 8 billion parameters, making them more resource efficient than larger models.

Despite these advancements, there is still a significant need to investigate the efficiency of LLM inference to ensure their environmentally friendly use. Researchers commonly use metrics such as time to first token and normalized energy consumption to gauge performance and energy efficiency of different models across various hardware settings [St24]. Wilkins et al. [WKM24] utilize metrics such as total energy consumption, runtime, energy per token, and throughput to evaluate and optimize the energy efficiency of LLM inference tasks across different hardware configurations. Faiz et al. [Fa24] evaluate the inference stage of LLMs by measuring the operational carbon footprint, total energy consumption, and hardware efficiency. They use these metrics to analyze the energy impact of various hardware configurations and data center efficiencies, providing a comprehensive assessment of the environmental footprint during the inference phase. Samsi et al. [Sa23] conducted benchmarks on LLMs using different GPUs to understand inference performance and energy costs. They made experiments to benchmark the inference performance and energy costs of different sizes of models using various metrics, including energy per second, energy per decoded token, energy per response, and the effects of GPU power capping to provide comprehensive understanding of energy efficiency of LLM inference. Argerich et al. [AP24] evaluate energy efficiency using metrics such as total energy consumption, energy consumption per component, energy efficiency, and inference latency across various LLMs. While current works present a variety of metrics for assessing the energy consumption, the need for comprehensive metrics that quantify the efficiency of LLM inference especially on edge devices is often overlooked. Our research aims to address this gap by focusing on efficiency evaluation on edge devices by using quality, runtime and power consumption as most important factors of influence.

In addition to inference, model training is particularly energy-intensive. Training of Llama 3 8B [Me24] emitted 390 tons of CO_2 equivalents which corresponds to 126 round-trip transcontinental flights from Berlin to New York for a single individual[3]. Additionally the training process required 1,3 million GPU hours. This amount of time equates to the entire lifetimes of 2 humans with a life expectancy of 74 years. Llama 3 70B, which is the largest version of the series, emitted 1900 tons of CO_2 equivalents and used 6,4 million GPU hours, emitting 4,87 times more CO_2 equivalents and using 5,3 more GPU-hours than Llama 3 8B.

[3] UBA Carbon Calculator, https://uba.co2-rechner.de/en_GB/mobility-flight, accessed: 06/24/2024

3 Methodology

The purpose of this work is to find out how we can use LLMs in an environmentally friendly way by examining their efficiency during inference on the edge. Therefore we envisioned a scenario where a curious user interacts with a conversational agent on an edge device, posing 50 distinct questions covering 33 different topics, each question taking 3 seconds to ask. The user asks no follow-up questions and only utilizes the questions from our dataset. In our scenario, the user repeats this process for three different LLMs, while quality, latency and energy demand is measured and recorded. The components of our experimental setup are described in this chapter.

3.1 Tools

All experiments in this work are conducted on a single-board computer. Specifically, we choose a Raspberry Pi 5 Model B Rev 1.0[4] as our edge device under test, suitable for a wide range of use cases ranging from educational to industrial applications. It is equipped with a 64-bit quad-core Arm Cortex-A76 processor running at 2,4GHz and 8GB of RAM. The single-board computer runs Raspberry Pi OS (64-bit), also known as Debian GNU/Linux 12 (bookworm). Energy measurements are performed by a USB-C multimeter[5] with ± 1‰ measurement accuracy. During measurements, it is connected to the single-board computer via USB-C/Power port and a laptop via USB port for data-logging purposes. Our conversational agent is based on Ollama[6], which we use to run different LLMs on our device under test. Additionally, Ollama Python Library is used to automate conversations.

3.2 Data

The questions we ask our conversational agent during the experiments originate from the Stanford Question Answering Dataset SQuAD2.0 [Ra16], which was created to understand the types of reasoning required by computer systems to answer these questions. We created a subset with 50 randomly selected questions and corresponding answers. The selected questions are about 33 different topics, including categories like Computational Complexity Theory, Geology, Oxygen, Imperialism and European Union Law. This is an example:

```
Topic: Prime Numbers
Question: What theorem defines the main role of primes in number theory?
Answer: The fundamental theorem of arithmetic
```

[4] RPi 5 Specs, https://datasheets.raspberrypi.com/rpi5/raspberry-pi-5-product-brief.pdf, accessed: 07/21/2024

[5] USB-C Multimeter Specs, https://joy-it.net/en/products/JT-TC66C, accessed: 07/21/2024

[6] Ollama Repository, https://github.com/ollama, accessed: 07/21/2024

3.3 Models

In our experiments, we investigate three distinct models designed to generate human-like text in response to text-based questions from users. The first model is Llama 3 8B [Me23], developed by Meta AI, featuring 8 billion parameters. It was pretrained on approximately 15 trillion tokens sourced from public data. The second model is Mistral 7B [Ji23] by Mistral AI, utilizing 7,3 billion parameters. It surpasses Llama 2 13B on all tested benchmarks and outperforms Llama 1 34B on several benchmarks. Mistral 7B focuses on achieving top-tier performance while ensuring efficient inference, according to its creators. However, details regarding the amount of training data and carbon emissions during its training phase are undisclosed. The third model is Phi-3 [Ab24], a 3,8 billion parameter language model by Microsoft, designed for deployment on edge devices like smartphones. It was trained on 3,3 trillion tokens, incorporating filtered web data from various public internet sources and synthetic data generated by other LLMs. Phi-3 is categorized as a Small Language Model (SLM), tailored for efficient inference tasks.

model	parameters	size	training data	quantization	carbon emitted
Llama 3	8B	4,7 GB	~15T tokens	4-bit	390 $tCO_2eq.$
Mistral	7,3B	4,1 GB	-	4-bit	-
Phi-3	3,8B	2,3 GB	3.3T tokens	4-bit	-

Tab. 1: Models which are used and compared in this work

4 Results

4.1 Quality

An important factor for evaluating the efficiency of a generative language model is the quality of its responses. A model that provides factually correct answers to various questions works more efficiently than a model producing mostly incomplete answers containing irrelevant information. Therefore, we evaluate the quality of the generated answers compared to the reference answers in the dataset by performing human-based and LLM-based quality analysis. Human-based evaluation is done by the authors, while LLM-based quality analysis is performed by GPT-4o[7]. We choose three weighted factors and set their values for evaluating our LLM-generated answers which are *accuracy* with $w_a = 0,5$, *completeness* with $w_c = 0,3$ and *relevance* with $w_r = 0,2$. We weighted *accuracy* higher than *completeness* and *relevance* because it affects the response's quality the most.

$$Q_I = \frac{1}{w_a a + w_c c + w_r r} \tag{1}$$

[7] GPT-4o, https://openai.com/index/hello-gpt-4o, accessed: 07/22/2024

Figure 1 demonstrates varying levels of answer quality among the models, while Mistral 7B performs best on our dataset, followed by Phi-3 3,8B and Llama 3 8B with quality scores ranging from 0,733 to 0,859. These values indicate that all models were able to answer the questions from the dataset with good to very good quality. The authors rate the quality to be slightly worse than GPT-4o for all models. This result illustrates, that good results are also possible with comparatively small models running on a heavily performance-limited device.

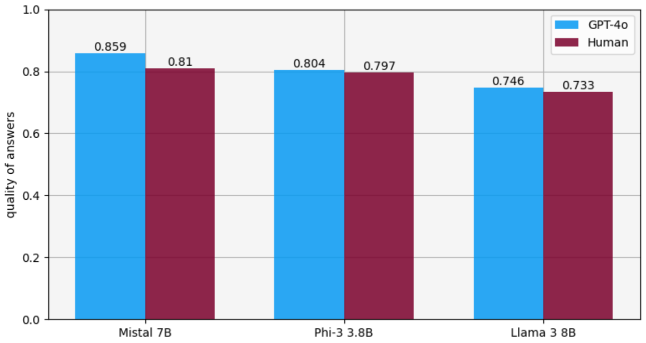


Fig. 1: GPT-4o and human-based quality evaluation of LLM-generated answers

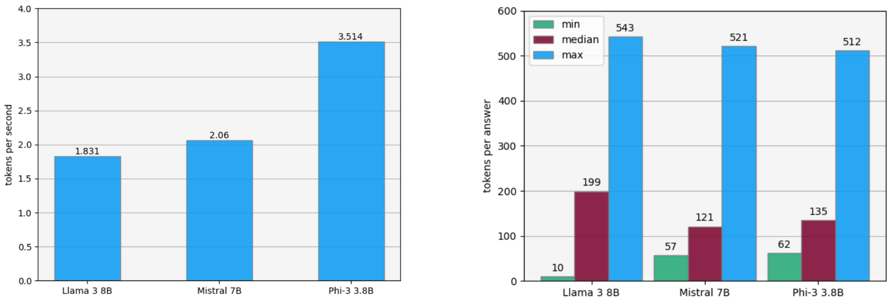
4.2 Time

Another important variable influencing efficiency is work done per unit of time. In our case, this is measured by the time it takes, to generate a single token, which corresponds to approx. 4 characters of common English text[8]. We developed the following metric to normalize the values between 0 and 1, where t_I corresponds to time per predicted token. The more tokens predicted per second, the lower the calculated value for t_I and the higher the value for T_I.

$$T_I = \frac{1}{1 + t_I} \tag{2}$$

Figure 2a reveals large differences in tokens generated per second. While Llama 3 8B generates 1,83 tokens, Phi-3 3,8B generates 3,51 tokens. Therefore, it is 1,92 times faster than Llama 3 8B and 1,70 times faster than Mistral 7B. The values in figure 2b reflect that Llama 3B has the largest median and max number of tokens generated per answer. While answering every question in the dataset, Llama 3 8B generated a total of 10.377 tokens, followed by Mistral 7B generating 8.779 tokens and Phi-3 3,8B generating 8.723 tokens in total.

[8] OpenAI Tokenizer, https://platform.openai.com/tokenizer, accessed: 07/22/2024



(a) Generated tokens per second

(b) Generated tokens per answer

Fig. 2: Comparison of generated tokens per second (a) and generated tokens per answer (b)

When considering these results from the perspective of efficiency, Phi-3 3,8B operates most efficient on our dataset, as it generates the most tokens per second and also produces the fewest tokens in total. Additionally, the models also require time for the initial loading phase which takes place at the beginning of a conversation. Llama 3 8B requires 58,73 seconds, Mistral 7B requires 49,77 seconds and Phi-3 3,8B requires 34,09 seconds on our hardware under test to load the model.

4.3 Energy

The third variable we investigate is energy demand. Our measuring device has a measurement accuracy of ± 1‰ and provides us with one value per second for volts (V) and amperes (A), which we use to determine the energy demand of our scenario in milliwatt-hours (mWh). Therefore we developed another metric to examine how much energy in mWh it takes to generate a single token. We added w_E as additional weight to make E_I adjustable.

$$E_I = \frac{1}{1 + w_E E_I} \tag{3}$$

Figures 3, 4 and 5 illustrate that the power consumption in watts for generating a response is approx. the same across all models. The baseline for the operating system is at 3,4 W on average. However, if we consider the entire time span it takes to answer every question in the dataset, it is evident that Phi-3 3,8B consumes the least amount of energy overall, because total energy demand is 8.146,94 mWh for Phi-3 3,8B, 13.338,30 mWh for Mistral 7B, and 18.230,88 mWh for Llama 3 8B.

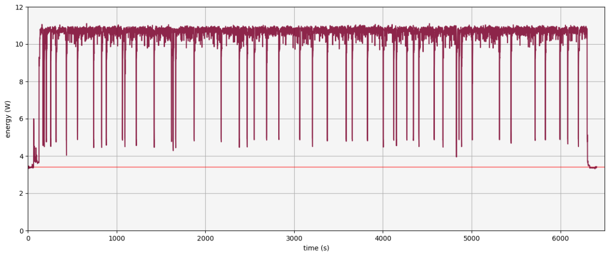


Fig. 3: Energy demand of Llama 3 8B to answer every question in the dataset

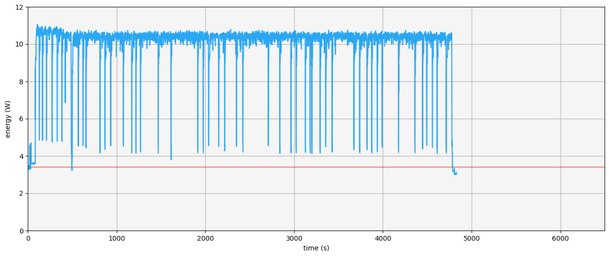


Fig. 4: Energy demand of Mistral 7B to answer every question in the dataset

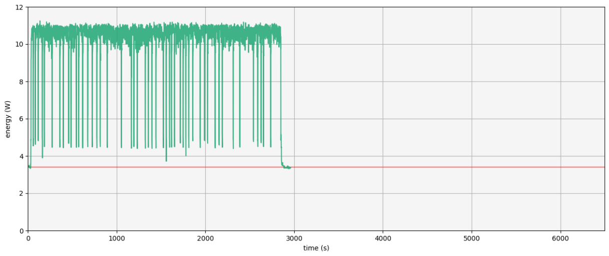


Fig. 5: Energy demand of Phi-3 3,8B to answer every question in the dataset

From an energy efficiency perspective, Phi-3 3,8B is most efficient in our scenario, as it has the lowest energy requirement overall with 0,93 mWh per token, followed by Mistral 7B.

model	per token	per question	total
Llama 3 8B	1,76 mWh	364,61 mWh	18.230,88 mWh
Mistral 7B	1,55 mWh	266,77 mWh	13.338,30 mWh
Phi-3 3,8B	0,93 mWh	162,94 mWh	8.146,94 mWh

Tab. 2: Energy demand of LLMs for answering 50 questions on an edge device

4.4 Efficiency

We formulate our final metric to assess the inference efficiency of LLMs on an edge device by incorporating our variables *quality*, *time* and *energy*. We weight each of the three factors with a value between 0 and 1, whereby the weights can be adjusted depending on individual or task related preferences. For the final evaluation, $w_E = 0,4$, $w_T = 0,3$ and $w_Q = 0,3$ are used as weights for the calculation of the values for our metrics *energy*, *quality*, *time* and the final efficiency score.

$$M_I = w_E E_I + w_T T_I + w_Q Q_I \tag{4}$$

According to the results in table 3 Mistral 7B generated the answers with the best overall quality, but finally, it is Phi-3 3,8B which has the best efficiency score in total.

model	energy	time	quality	total (M_I)
Phi-3 3,8B	0,380	0,759	0,804	0,621
Mistral 7B	0,092	0,673	0,859	0,496
LLama 3 8B	0,070	0,646	0,746	0,446

Tab. 3: Efficiency of the inference of LLMs on a resource-constrained edge device

5 Discussion

In this paper, we investigated the efficiency of LLM inference on edge devices and developed metrics to quantify it. We established three central components to measure inference efficiency: *quality* of the generated text, *time* for generating the text, and *energy demand* during token prediction. The quality of answers was assessed by a human-based evaluation performed by the authors and an LLM-based evaluation performed by GPT-4o. Quality evaluation was done by rating the generated answers to 50 different questions based on the factors *accuracy*, *completeness*, and *relevance*. The models under test achieved quality scores ranging from 73,3% to 85,9% and thus demonstrate their ability to generate sufficient answers.

Each model achieved distinct speed in token generation, ranging from 1,831 tokens per second for Llama 3 8B, to 2,06 tokens per second for Mistral 7B, and up to 3,514 tokens per second for Phi-3 3,8B. These values indicate, that the models are definitely suitable for practical use on a device with very limited ressources. However, it should also be taken into account that loading the model into memory can also take a significant amount of time. Additionally, Llama 3 8B generated 10.377 tokens to answer all 50 questions in the dataset, whereas Mistral 7B created 8.779 tokens and Phi-3 3,8B produced 8.723 tokens. These values illustrate that the number of model parameters has a significant influence on the speed of inference, as Phi 3 3,8B was considerably faster than Llama 3 8B.

Energy demand was measured in milliwatt-hours (mWh), with Phi-3 3,8B having the lowest energy demand at 0,934 mWh per token and Llama 3 8B having the highest demand at 1,76 mWh per token. While Phi 3 3,8B required a total of 8.146,94 mWh to answer all questions, Mistral consumed 13.338,30 mWh and Llama 3 8B consumed 18.230,88 mWh. Finally, we quantified overall efficiency, finding that Phi-3 3,8B had the best efficiency score at 0,621, followed by Mistral 7B at 0,569, and Llama 3 8B at 0,504. This clearly shows that the model with the fewest parameters (Phi 3 3,8B) performs best on our dataset and is the most efficient LLM we tested.

6 Conclusion

Our research demonstrates that relatively small state of the art large language models with 3,8 to 8 billion parameters can run on a modern single-board computer with very limited computational resources without the need for a GPU or external inference accelerators. Our findings show that a model with the most parameters does not necessarily achieve the best performance, nor is it most efficient. The ability to generate responses in reasonable time and with satisfactory quality indicates that these models are suited for practical use, which makes them useful for a wide range of text-based applications, ranging from voice assistants to text summarization tools. In future research, we plan to explore the efficiency of various LLMs on different hardware. While this study focused on hardware with limited resources, we aim to compare hardware specifically designed for AI applications in the future. It is also conceivable to explore efficiency optimization opportunities of LLM-based systems on edge devices based on the results of this work.

In conclusion, our study underscores the importance of considering not only the performance metrics of LLMs on edge devices but also their energy efficiency and overall suitability for practical applications requiring reasonably fast responses while demanding only small amounts of energy. However, the user should note that the conversational agent based on the models examined in this work is a seemingly competent generalist, rather than a narrow specialist in every possible field. Therefore its answers should always be questioned.

Acknowledgements

This work was funded by the German Federal Ministry for Economic Affairs and Climate Action as part of the project *EASY* under grant No. 01MD22002C and by the German Federal Ministry for the Environment, Nature Conservation, Nuclear Safety, and Consumer Protection as part of the project *KIRA* under grant No. 67KI32013B.

Parts of the text have been enhanced and linguistically revised using AI tools. All concepts and implementations described are the intellectual work of the authors.

References

[Ab24] Abdin, M. et al.: Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone, 2024, arXiv: 2404.14219, visited on: 07/25/2024.

[AP24] Argerich, M. F.; Patiño-Martínez, M.: Measuring and Improving the Energy Efficiency of Large Language Models Inference. IEEE Access 12, 2024.

[Fa24] Faiz, A. et al.: LLMCarbon: Modeling the end-to-end Carbon Footprint of Large Language Models, 2024, arXiv: 2309.14393, visited on: 07/25/2024.

[Ji23] Jiang, A. Q. et al.: Mistral 7B, 2023, arXiv: 2310.06825, visited on: 07/25/2024.

[Li24] Li, L. et al.: Norm tweaking: High-performance low-bit quantization of large language models. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 38 17. 17, pp. 18536–18544, 2024.

[Me23] Meta: Introducing Meta Llama 3: The most capable openly available LLM to date, 2023, URL: https://ai.meta.com/blog/meta-llama-3, visited on: 03/20/2024.

[Me24] Meta: Llama 3 Model Card, 2024, URL: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md, visited on: 04/03/2024.

[Op24] OpenAI: GPT-4 Technical Report, 2024, arXiv: 2303.08774, visited on: 07/25/2024.

[Ra16] Rajpurkar, P. et al.: SQuAD: 100,000+ Questions for Machine Comprehension of Text, 2016, URL: https://nlp.stanford.edu/pubs/rajpurkar2016squad.pdf, visited on: 07/25/2024.

[Sa23] Samsi, S. et al.: From Words to Watts: Benchmarking the Energy Costs of Large Language Model Inference. In: IEEE High Performance Extreme Computing Conference (HPEC). Vol. 1, pp. 1–9, 2023.

[St24] Stojkovic, J. et al.: Towards Greener LLMs: Bringing Energy-Efficiency to the Forefront of LLM Inference, 2024, arXiv: 2403.20306, visited on: 04/03/2024.

[Su24] Sun, M. et al.: A Simple and Effective Pruning Approach for Large Language Models, 2024, arXiv: 2306.11695, visited on: 07/25/2024.

[Va17] Vaswani, A. et al.: Attention is All you Need. In: Advances in Neural Information Processing Systems. Vol. 30, pp. 5998–6008, 2017.

[WKM24] Wilkins, G.; Keshav, S.; Mortier, R.: Hybrid Heterogeneous Clusters Can Lower the Energy Consumption of LLM Inference Workloads. In: ACM International Conference on Future and Sustainable Energy Systems. Vol. 15, pp. 506–513, 2024.