Green AI by Default: Energy Reduction Techniques for LLMs in SE

Enrique Barba Roque*
e.barbaroque@tudelft.nl
Delft University of Technology
Delft, The Netherlands

Abstract

Large Language Models (LLMs) are increasingly being applied to Software Engineering (SE) tasks, showing high accuracy in various problems. However, their high computational demands and energy consumption raise sustainability concerns and hinder their use on consumer hardware and resource-constrained platforms. Multiple optimization techniques exist, but they are often neglected due to the technical difficulty of applying them during model development. This research aims to improve the accessibility of optimization techniques by (1) making energy reporting of LLM more accessible, (2) streamlining and automating optimization techniques, (3) providing guidelines to select appropriate techniques for different use cases, and (4) exploiting these techniques to design efficient SLM ensemble architectures for LLM-enabled applications. Expected contributions include methods for measuring and reporting energy usage, tools to automate compression of models, guidelines to guide developers through the optimization process, and using these results to explore alternative deployment setups for compressed LLMs.

CCS Concepts

• **Computing methodologies** → **Natural language generation**; • **Hardware** → **Impact on the environment**; • **Software and its engineering** → Application specific development environments.

Keywords

Green AI, LLM Compression, LLMs for Code, LLM Energy Usage, AI for SE

ACM Reference Format:
Enrique Barba Roque. 2026. Green AI by Default: Energy Reduction Techniques for LLMs in SE. In *2026 IEEE/ACM 48th International Conference on Software Engineering (ICSE-Companion '26), April 12–18, 2026, Rio de Janeiro, Brazil.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3774748.3787655

1 Introduction and Related Work

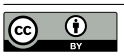
AI models have taken the front stage with respect to research and innovation. The main paradigm is Large Language Models (LLMs), which have emerged as a highly useful tool for tasks that require language understanding and generation, including Software Engineering (SE) tasks [7]. Their capabilities arise from their scale: billions of parameters trained on terabytes of open-source code from platforms such as GitHub [15]. This scale brings significant computational challenges, including the need for dedicated hardware for parallel computation, such as GPUs [23], along with high energy demands and environmental costs [19]. The International Energy Agency projects that the energy consumption of AI is expected to double in the next 5 years [12], further straining existing electrical infrastructure and increasing greenhouse gas emissions.

Optimizing the energy usage of LLMs requires considering multiple aspects. First, although training is more computationally expensive, inference often dominates lifetime costs for widely used models with long life cycles and many queries [6]. Second, general-purpose models tend to be more energy intensive than task-specific ones, especially as input and output sequences grow longer [16]. Finally, for SE applications, compressed models are preferable, since they can be deployed on consumer hardware within an IDE, rather than relying on remote server infrastructure [17, 21].

Smaller language models can also be part of novel architectures for AI-enabled systems. Recent research directions on LLMs are moving towards substituting large models with a combination of smaller models. A recent position paper [2] argues that Small Language Models (SLMs) are the future of agentic AI, where large generic agents can be substituted by an ensemble of smaller, task-specific SLMs, leading to better energy efficiency and shorter inference times, among other advantages. A similar approach is present in recent works for LLM routing [10, 8, 22]. These methods introduce a predictive model that routes a prompt to one or more LLMs out of a selection of models. To do this. The router analyzes a given prompt, predicts accuracy and costs for the available models, and tries to balance both objectives. However, most of the current approaches focus only on costs as API monetary costs, rather than energy usage.

There are multiple well-known techniques to compress large models into smaller ones, such as quantization, model pruning, or knowledge distillation [9]. In the current development pipeline for LLMs, optimization is not at the forefront. Applying compression techniques is a time-consuming task. Different techniques can have different implementations depending on the model and tasks, selecting the configuration parameters is a very manual task, and APIs are not standardized. On top of that, it is difficult to know beforehand the impact that a compression technique will have on the accuracy and energy consumption of a model. Making compression techniques easier to apply can facilitate the work of developers and open up the possibility of reduced AI energy consumption across the industry, thanks to the increased accessibility. Accessible compression techniques can also support newer SLM architecture paradigms by facilitating the compression necessary for these models. In the context of SE tasks, some papers try to tackle part of this problem. One such example is modeling the configuration parameters as a Many-Objective Optimization Problem (MOP), with accuracy and computational costs as objectives [17, 20]. However,

*Year 2 out of expected 5. Supervisor: Luis Cruz



ICSE-Companion '26, Rio de Janeiro, Brazil

ACM ISBN 979-8-4007-2296-7/2026/04
https://doi.org/10.1145/3774748.3787655

these approaches use Floating Point Operations (FLOPs) as the objective metric for computational cost, for which its correlation with energy consumption is not clear [6, 1].

The objective of this project is to make these insights accessible to developers. First, we want to facilitate the energy assessment of LLM models by building reliable metrics and estimations, so they can avoid cumbersome energy measurements. Second, we aim to provide pre-assessment methods so the AI developer can judge if the costs of applying a compression technique are worth the trouble, depending on the lifetime of the model. Third, we will look into easing the application of compression techniques by providing a framework tool that can automatically apply optimal parameters for the different techniques. Using this framework, we will perform a large-scale study to measure the accuracy and energy impact of different compression techniques, and translate this into a set of guidelines per architecture and task that help developers select the best technique for their final goal. Finally, we will validate the usefulness of these results by using them to explore LLM Routing architectures, using the framework to create LLM ensembles focusing on energy efficiency.

2 Research Questions

To bring energy efficiency of LLMs to the foreground, we need to facilitate the compression and optimization process for models and find alternative architectures for LLM-enabled applications, based on SLM ensembles with a lower energy consumption. To achieve this, we identify the following problems to tackle: (1) energy usage of LLM models has to be accessible and easier to measure, and standardized as a metric in benchmarks alongside accuracy, (2) model compression should be streamlined, developing and relying on automated approaches such as MOP to determine optimum parameters and tradeoffs, (3) provide guidelines to developers on which compression techniques to applied based on the original model and task to perform, and (4) exploit these techniques to design efficient SLM ensemble architectures for LLM-enabled applications. To do this, we propose the following research questions:

RQ1. *What is an appropriate energy proxy metric that is accurate, easy to measure, and provides actionable information to the developer?*

RQ2. *How should efficiency gains of compressed LLM be reported?*

RQ3. *What methods can be exploited to develop automated approaches to model compression?*

RQ4. *Which compression methods are the most effective in terms of efficiency, given a base model and an SE task?*

RQ5. *How can compressed LLMs be used to design energy-efficient SLM ensembles?*

3 Expected Contributions

To tackle the Research Questions presented in this proposal, we expect to make four contributions. For each of these contributions, we aim to publish a peer-reviewed paper and a replication artifact with source code.

FLOPs Validity as Energy Proxy. In present literature, FLOPs is commonly used as a proxy metric when reporting energy costs of an LLM [14, 13]. However, the validity of this metric as a proxy of energy is still an open discussion in the research community [1, 5, 4]. We aim to test the inference energy of different LLMs and validate whether minimizing FLOPs in MOP also minimizes energy consumption. To evaluate this, we apply the state-of-the-art MOP distillation technique [17] for a selection of SE classification and generation tasks, measuring energy consumption through software profilers. Then, we check for correlation between FLOPs and actual energy measurements. With these results, we contribute to **RQ1**.

Accurate Energy Proxies. An advantage of using FLOPs as a cost metric is that they are easy to compute and measure based on the architecture of the model [6]. On the other hand, using energy profilers is much more impractical. Since they read CPU and GPU registries, they require root access to the host OS and bare metal, so they cannot be used inside Virtual Machines or containers in the cloud. On top of that, there are a series of guidelines to follow to make sure measurements are not tainted by other processes in the same machine [3]. Therefore, it is relevant to find a metric that lies in the middle. This metric should correlate with actual energy consumption accurately while being easy to measure, for example, through a short inference run rather than longer energy measurements. As a starting point, we will look into the work of Asperti et al. [1], who proposes a correction for FLOPs for convolutional networks, and see if we can apply a similar method for transformer-based models. Evaluation will test at least one model over an SE benchmark, computing the energy proxy and comparing its accuracy and correlation with actual energy measurements. This project contributes to **RQ1** and **RQ2**.

Compression Amortization. Many compression techniques are not straightforward to apply. For example, knowledge distillation requires several steps. A suitable configuration for the student model needs to be identified. Then, the student model must be trained, which requires not only the training compute for the student but also running inference on the teacher. In exchange, the student model will run with cheaper inferences. If the final compressed model is only used for a small number of inferences, the compression cost might outweigh the savings. The objective is to provide methods to estimate the cost of compression and the cost of inference before applying a compression technique, and give an approximate amortization window. Evaluation will include at least one model and several techniques, including knowledge distillation and quantization. We will test accuracy and energy consumption for different SE Engineering tasks. For energy measurements, we aim to use the energy proxies from the previous contribution as well as actual energy experiments. These results contribute to **RQ2**.

Automated Model Compression. Compressing an AI model can be difficult, requiring certain knowledge of the techniques and a significant time investment. The objective is to provide an easy and automated way of compressing an AI model for a given task. The user provides the model to compress and a task dataset, and selects which techniques to apply. We will exploit MOP capabilities as done for knowledge distillation [17], extending its application to other compression techniques. This project will be a tool demonstration project, rather than a fully fledged research paper. To evaluate this framework, we will test several models and tasks, with at least one classification and one generation, and validate by reporting the accuracy and energy consumption of the compressed models, and their gains in efficiency. These results contribute to **RQ3**.

Technique effectiveness catalogue. Different SE tasks have different design requirements, and benefit better from certain architectures rather than others. For example, Code-to-Code tasks work better with decoder-only models. In the same way, certain tasks

or architectures can benefit differently from different compression techniques. The objective of this project is to leverage the previous framework to perform a large-scale study and identify the impact of different compression techniques on accuracy and energy consumption for different tasks. The final product of this project would be a set of guidelines, companion to the framework, recommending the best performing techniques for the different tested tasks. Evaluation will consist of a series of tasks and benchmarks, and define reference models that perform the best on those tasks. Then, we will test different compression techniques, and measure accuracy loss and energy savings, cataloging the impact for each task and technique. With these results, we contribute to **RQ4**.

Orchestration of Small Language Models. The concept of SLM Orchestration proposes an architectural framework that dynamically routes prompts between a Large Language Model (LLM) and an ensemble of Small Language Models (SLMs) to optimize both accuracy and energy efficiency. The router is a predictive model, which can also be a small language model, that predicts the performance and energy usage that each of the SLMs in the ensemble will have on a given prompt. This approach operates along two main dimensions: first, by leveraging SLMs with varying accuracy levels to adapt resource usage based on task requirements; and second, by employing expert SLMs trained or compressed from a larger LLM to specialize in specific tasks. The goal is to achieve comparable or improved performance while significantly reducing energy consumption. To evaluate this, we will select some SE generation and classification tasks, and report performance and energy savings achieved through this orchestration compared to using a single LLM. We will also report and study the overhead introduced by the routing mechanism. This project contributes to answering **RQ5**.

Alternative Foundational Architectures. As a small part of our research, we want to look into how newer AI alternatives to transformers can be applied to SE tasks. Concretely, we want to explore the use of Spiking Neural Networks (SNNs). In SNNs, information is encoded in binary spikes across time, which brings large energy savings compared to traditional neural networks [11]. However, SNNs require specific neuromorphic hardware to achieve these gains, which is still not widely available, and difficult the optimization of these networks [18]. By exploring their applications for SE tasks, we pave the way for the future, when neuromorphic hardware becomes more accessible.

4 Initial Results

To date, we are in the process of publishing the first contribution. We tested the validity of FLOPs as an energy proxy for MOP minimization during knowledge distillation. We found that the correlation between FLOPs and energy is often unreliable, and using MOP to optimize for lower FLOPs does not always lead to lower energy consumption. We proposed an alternative approach, estimating the energy consumption of a set of hyperparameters directly through surrogate modelling, which leads to better results in terms of energy consumption.

References

[1] Andrea Asperti, Davide Evangelista, and Moreno Marzolla. 2022. Dissecting flops along input dimensions for greenai cost estimations. In *Machine Learning, Optimization, and Data Science*. Springer International Publishing, Cham, 86–100. ISBN: 978-3-030-95470-3.

[2] Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. 2025. Small language models are the future of agentic ai. (2025). arXiv: 2506.02153 [cs.AI]. doi:10.48550/arXiv.2506.02153.

[3] Luís Cruz. 2021. Green software engineering done right: a scientific guide to set up energy efficiency experiments. Retrieved May 27, 2024 from http://luiscruz.github.io/2021/10/10/scientific-guide.html.

[4] Luís Cruz et al. 2025. Greening ai-enabled systems with software engineering: a research agenda for environmentally sustainable ai practices. *SIGSOFT Softw. Eng. Notes*, 50, 3, (July 2025), 14–23. doi:10.1145/3743095.3743099.

[5] Santiago del Rey, Silverio Martínez-Fernández, Luís Cruz, and Xavier Franch. 2023. Do dl models and training environments have an impact on energy consumption? In *2023 49th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. (Sept. 2023), 150–158. doi:10.1109/SEAA60479.2023.00031.

[6] Radosvet Desislavov, Fernando Martínez-Plumed, and José Hernández-Orallo. 2023. Trends in ai inference energy consumption: beyond the performance-vs-parameter laws of deep learning. *Sustainable Computing: Informatics and Systems*, 38, 100857. doi:10.1016/j.suscom.2023.100857.

[7] Angela Fan, Beliz Gokkaya, Mark Harman, Mitya Lyubarskiy, et al. 2023. Large language models for software engineering: survey and open problems. In *2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE)*. IEEE, 31–53.

[8] Tao Feng, Yanzhen Shen, and Jiaxuan You. 2024. Graphrouter: a graph-based router for llm selections. *arXiv preprint arXiv:2410.03834*.

[9] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. (2015). arXiv: 1503.02531 [stat.ML].

[10] Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. 2024. Routerbench: a benchmark for multi-llm routing system. *arXiv preprint arXiv:2403.12031*.

[11] Paul Hueber, Guangzhi Tang, Manolis Sifalakis, Hua-Peng Liaw, et al. 2024. Benchmarking of hardware-efficient real-time neural decoding in brain–computer interfaces. *Neuromorphic Computing and Engineering*, 4, 2, 024008.

[12] International Energy Agency. 2025. Energy and AI. Special Report. Published April 2025. CC BY 4.0 licence. International Energy Agency (IEA), Paris, France, (Apr. 2025). https://www.iea.org/reports/energy-and-ai.

[13] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, et al. 2020. TinyBERT: distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, (Nov. 2020), 4163–4174. doi:10.18653/v1/2020.findings-emnlp.372.

[14] Xiangyang Liu, Tianxiang Sun, Junliang He, Jiawen Wu, et al. 2022. Towards efficient NLP: A standard evaluation and A strong baseline. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022*. doi:10.18653/V1/2022.NAACL-MAIN.240.

[15] Anton Lozhkov, Raymond Li, Loubna Ben Allal, et al. 2024. Starcoder 2 and the stack v2: the next generation. (2024). arXiv: 2402.19173 [cs.SE].

[16] 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* (FAccT '24). Brazil, 85–99. ISBN: 9798400704505. doi:10.1145/3630106.3658542.

[17] Annibale Panichella. 2025. Metamorphic-based many-objective distillation of llms for code-related tasks. In *ICSE 2025*. doi:10.1109/ICSE55347.2025.00230.

[18] Enrique Barba Roque and Luís Cruz. 2025. Energy aware development of neuromorphic implantables: from metrics to action. *CoRR*, abs/2506.09599. arXiv: 2506.09599. doi:10.48550/ARXIV.2506.09599.

[19] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020. Green ai. *Commun. ACM*, 63, 12, (Nov. 2020), 54–63. doi:10.1145/3381831.

[20] Jieke Shi, Zhou Yang, Hong Jin Kang, Bowen Xu, et al. 2024. Greening large language models of code. In *Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Society* (ICSE-SEIS'24). Association for Computing Machinery, Lisbon, Portugal, 142–153. ISBN: 9798400704994. doi:10.1145/3639475.3640097.

[21] Alexey Svyatkovskiy, Sebastian Lee, Anna Hadjitofi, Maik Riechert, et al. 2021. Fast and memory-efficient neural code completion. In *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)*, 329–340. doi:10.1109/MSR52588.2021.00045.

[22] Xinyuan Wang, Yanchi Liu, Wei Cheng, Xujiang Zhao, Zhengzhang Chen, Wenchao Yu, Yanjie Fu, and Haifeng Chen. 2025. Mixllm: dynamic routing in mixed large language models. *arXiv preprint arXiv:2502.18482*.

[23] Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, et al. 2022. Sustainable AI: environmental implications, challenges and opportunities. In *Proceedings of the Fifth Conference on Machine Learning and Systems, MLSys 2022, Santa Clara, CA, USA, August 29 - September 1, 2022*. mlsys.org.

Received January 8, 2026