EVALUATING THE ENVIRONMENTAL IMPACT OF LANGUAGE MODELS WITH LIFE CYCLE ASSESSMENT

Jared Fernandez*, Clara Na*, Yonatan Bisk, Emma Strubell
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
jaredfern@cmu.edu

1 INTRODUCTION

The development and use of AI models is an increasingly complex ecosystem with multiple software and hardware components deployed over several stages, each with non-negligible environmental costs that contribute to total carbon emissions and environmental impact (Gupta et al., 2021). As these machine AI models have grown in scale, so have the energy demands and the resulting environmental impact of development and deployment (Strubell et al., 2020; Luccioni et al., 2024).

The pursuit of improved model capabilities has resulted in rapid scaling of model training and inference, while neglecting the environmental consequences – carbon emissions from internet and communication technologies growing at rates that far exceed those of other sectors (Knowles et al., 2022). To support the scale necessary for the industrial development of large machine learning models, government and industry institutions put forward large capital expenditure on data center hardware, and energy infrastructure to support large-scale computing facilities (Parashar et al., 2023). These investments yield commensurate increases in the energy demand, water use, associated carbon emissions costs as the increased scale, prevalence, and accessibility of machine learning models produces higher utilization; with projections estimating that data centers will consume between 9.1% and 11.7% of the total US energy demand by 2030 (Aljbour et al., 2024); and up to 6.6 billion cubic meters of water worldwide by 2027 (Li et al., 2022).

In response, recent research in efficient machine learning and Green AI has proposed interventions aimed at reducing the environmental resource consumption of machine learning (Schwartz et al., 2020). Additionally, various tools and frameworks have facilitated reporting and measurement of metrics related to efficiency and environmental impact, and it is increasingly common for institutions developing models to report the energy cost of the final training run of large models (Zhang et al., 2022; Dubey et al., 2024).

Unfortunately, total operational costs of development and inference deployment remain poorly characterized. Existing efficiency research remains siloed and modular, and sector-wide projections are insufficient in granularity to fully characterize the growing environmental impact of the development and deployment of individual machine learning models. In order to account for the environmental impact over the full life of a model it is necessary to account for both the embodied emissions from the manufacture of computing hardware and construction of facilities, and the operational emissions over all stages of development and deployment of models.

To provide a comprehensive understanding of the environmental impacts of machine learning development and deployment, we propose applying life cycle assessment to analyze multi-stage training and inference with state-of-the-art large language models.

2 PROPOSAL: LARGE LANGUAGE MODEL LIFE CYCLE ANALYSIS

We propose applying life cycle assessment LCA, a commonly used methodology in environmental ecology (Curran, 1996; 2006) to analyze the development and deployment life cycle of machine learning models; specifically, the life cycle of large language models. In life cycle assessment, the life of a product or model is decomposed into component stages over the course of its manufacture, use, and construction to estimate the total impact of aggregation over its full lifetime.

For machine learning models, this includes the embodied carbon from manufacture of hardware and construction of data centers; and the operational carbon emitted due to energy used during the development and deployment of models. Although previous work characterizing stages of the model life cycle (Wu et al., 2022; Morrison et al., 2025; Luccioni et al., 2023; 2024) provides insight into individual portions of the model life training and inference life cycle, they are not reflective of modern language model development pipelines which have grown in complexity to include multistage pre- and post-training, sometimes with additional specific domain adaptation; as well as inference-time methods with greater computational intensity than traditional inference (e.g. chain-of-thought reasoning and in-context learning).

We will conduct this research in two phases: (1) empirical benchmarking of the operational carbon from previously unstudied stages of the model life cycle; (2) investigation of the tradeoffs between shifting computational load and environmental impact between stages.

2.1 Phase 1: Methods for Model Life Cycle Analysis

We begin by providing proposals to characterize the energy requirements of machine learning use in development and deployment settings; we then characterize the energy use across stages of the machine learning life cycle and the corresponding hardware platforms.

In contrast to prior work that solely studies machine learning energy use as attributed to GPU power draw, we will measure the power draw for hardware components required in machine learning systems (i.e. across GPU hardware accelerator, CPU, and memory). We will then quantify the carbon dioxide equivalent emissions and energy consumption on a per-component basis.

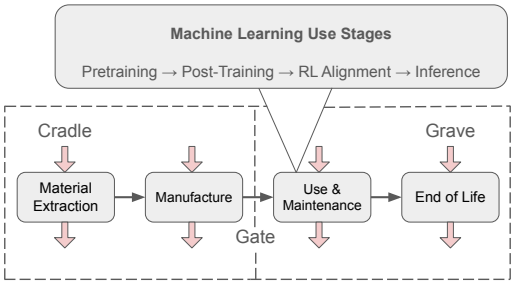

Figure 1: Stages of Language Model Life Cycle Assessment. Growing in complexity over time, the use stage of machine learning models has grown to include many components which each contribute to overall operational emissions.

Next, we define a set of common settings for training development and inference deployment based on the best practices of modern open language models (Lambert et al., 2024; Dubey et al., 2024) to analyze as stages in life cycle assessment. Specifically, we apply our described energy measurements to conduct an expanded study in the pre-training, instruction tuning post-training, and reinforcement learning stages of language model development (see Figure 1). To model representative inference settings, we will estimate the expected serving loads for production language models performing offline and online serving with variable sequence lengths and batch sizes; and apply our described energy assessments to establish projections for the environmental impact of deployment. To understand the impacts of hardware and software design choices at each stage, we will examine representative software frameworks and inference engines (e.g. PyTorch, vLLM) as well as datacenter and workstation accelerator hardware (e.g. A6000, A100 GPUs) for both training and inference.

Based on the measured power use and carbon equivalents at each use stage, we will determine total operational carbon by aggregating across each stage of development and deployment; and determine the total carbon by accounting for the embodied carbon using approximations established in (Wu et al., 2022; Luccioni et al., 2023; Gupta et al., 2021).

2.2 Phase 2: Understanding trade-offs and interdependence

Beyond thorough accounting, we posit that life cycle analysis allows for improved understanding of the *interactions* and *trade-offs* between different stages of the machine learning life cycle. We will use the energy accounting described in §2.1 to understand the absolute costs of different stages, and characterize their relative interactions via breakeven points; to determine the best allocation of resources across stages.

Pre-train, then fine-tune The modern paradigm of "pre-train then fine-tune," refers to a practice where a single organization pre-trains a large model with a self-supervised learning objective

and the model is subsequently fine-tuned many times using less compute for many downstream use cases. This relies on an assumption that the large upfront costs of building the pre-trained model are made worthwhile by the resulting reduction in costs of subsequent downstream adaptation. This brings with it notions of "breakeven" points. One natural consideration is the point at which total cost savings in fine-tuning reach the initial cost of pre-training: $\texttt{cost}(PT|rand\_init) = \sum_{i \in FT} \texttt{cost}(FT_i|rand\_init) - \texttt{cost}(FT_i|PT)$.

In practice, it is difficult to empirically quantify $\texttt{cost}(FT_i|rand\_init)$, especially as it is commonly assumed that $\texttt{cost}(FT_i|PT) << \texttt{cost}(FT_i|rand\_init)$, and it is commonly assumed that the condition is met fairly quickly. A different, more measurable notion of breakeven point is sometimes considered: $\texttt{cost}(PT|rand\_init) = \sum_{i \in FT} \texttt{cost}(FT_i|PT)$. In this version, the total costs of fine-tuning meet the costs of pre-training, which may be interpreted as a signal of the pre-training cost being "worth" it, but, considered in aggregate over multiple models, can also be a way to quantify the relative computational demands of pre-training and fine-tuning over time. We propose to conduct a survey on self-supervised autoregressive pretraining versus fine-tuning breakeven points to determine the energy efficiency and effectiveness of additional pretraining as opposed to fine-tuning under a fixed energy or compute budget.

Train-time vs. inference-time compute Efficiency in AI systems is often considered under a performance constraint (e.g. a desired accuracy), whether in the form of an explicit trade-off *between* efficiency and accuracy or a fixed required accuracy level constraining the space of feasible efficiency interventions. Thus, in practice, increased computation performed at one stage of the model life cycle may decrease computational work needed at another stage, or vice versa. Concretely, supervised fine-tuning (SFT) or reward modeling can be used in post-training of language models to achieve desired behavior or performance on a specific task or general interactions between model and users. However, it is also possible in many cases to achieve desired behavior through prompting (e.g. "`<paragraph to summarize>. tl;dr:`" or in-context learning (ICL: e.g. prepending eight examples of text style transfer to the example(s) we want to perform style transfer on). We propose an empirical study of explicit post-training, prompting, and in-context learning to determine their relative energy requirements and to determine breakeven points as before; in this context: e.g. $\texttt{cost}(SFT|PT) = \sum_{i \in INF} \texttt{cost}(INF_i|PT) - \sum_{i \in INF} \texttt{cost}(INF_i|SFT)$.

Alternatively, post-training can also be leveraged to induce behaviors that increase inference-time compute such as with additional generated tokens in "chain-of-thought reasoning" (Xu et al., 2025). We propose an analysis of the post-training and inference-time compute energy costs, across both the upfront post-training cost and the additional marginal cost of inference incurred over inference-time use from longer generations. Analysis will consider measured and projected demand, increases or lack thereof in generation quality, and operational inference-time costs incurred over a model life cycle.

3 METHODOLOGY

As described in §2, we plan to conduct a fine-grained life cycle assessment of large language models. Specifically, we will examine the OLMo and Llama families of transformer decoder-only models (OLMo et al., 2024; Dubey et al., 2024) as popular open-source models representative of modern large language models. Additionally, we will consider reasoning models such as from the DeepSeek R1 family (DeepSeek-AI et al., 2025) along with their non-reasoning variants.

For direct measures of environmental resource use, we focus on energy use by compute hardware and carbon dioxide emission equivalents (CO2e). We will estimate the energy utilization with CodeCarbon framework (Courty et al., 2024), which leverages Nvidia Management Library and Intel Running Average Power Limit to measure GPU, CPU, and RAM energy use.

For executing the pre-training versus finetune experiments in §2.2, we will impose fixed total compute for pretraining and finetuning; and investigate the performance of finetuning OLMo models from released intermediate pretraining checkpoints. To investigate the trade-offs between train-time and inference-time compute, we will examine a variety of popular NLP datasets for performing task-specific (e.g. IMDB; Maas et al. (2011)) supervised finetuning, instruction tuning (Lambert et al., 2025), and make empirical cost comparisons with in-context learning (Brown et al., 2020).

4 Expected Outcomes

Performing LCA of ML models enables evaluation of the complete environmental costs associated with both development and deployment of machine learning models. Such insights can provide policy makers and institutions with information to guide regulation and decisions around the use of machine learning models. Furthermore, we expect that accurate decomposition of the machine learning life cycle use can provide a basis for reasoning about the effects of tradeoffs between allocating computational resources across different stages; and for evaluating the impact of efficiency improvements in a single stage on the environmental impact of a model's overall life cycle.

References

Jordan Aljbour, Tom Wilson, and P Patel. Powering intelligence: Analyzing artificial intelligence and data center energy consumption. *EPRI White Paper no. 3002028905*, 2024.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Benoit Courty, Victor Schmidt, Sasha Luccioni, Goyal-Kamal, MarionCoutarel, Boris Feld, Jérémy Lecourt, LiamConnell, Amine Saboni, Inimaz, supatomic, Mathilde Léval, Luis Blanche, Alexis Cruveiller, ouminasara, Franklin Zhao, Aditya Joshi, Alexis Bogroff, Hugues de Lavoreille, Niko Laskaris, Edoardo Abati, Douglas Blank, Ziyao Wang, Armin Catovic, Marc Alencon, Michał Stechły, Christian Bauer, Lucas Otávio N. de Araújo, JPW, and MinervaBooks. mlco2/codecarbon: v2.4.1, May 2024. URL https://doi.org/10.5281/zenodo.11171501.

Mary Ann Curran. Environmental life-cycle assessment, 1996.

Mary Ann Curran. *Life-cycle assessment: principles and practice*. National Risk Management Research Laboratory, Office of Research and Development, U.S. Environmental Protection Agency, 2006.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou,

Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Udit Gupta, Young Geun Kim, Sylvia Lee, Jordan Tse, Hsien-Hsin S Lee, Gu-Yeon Wei, David Brooks, and Carole-Jean Wu. Chasing carbon: The elusive environmental footprint of computing. In *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 854–867. IEEE, 2021.

Bran Knowles, Kelly Widdicks, Gordon Blair, Mike Berners-Lee, and Adrian Friday. Our house is on fire: The climate emergency and computing's responsibility. *Communications of the ACM*, 65 (6):38–40, 2022.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. T\" ulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training, 2025. URL https://arxiv.org/abs/2411.15124.

Pengfei Li, Jianyi Yang, Mohammad A Islam, and Shaolei Ren. Making ai less "thirsty": Uncovering and addressing the secret water footprint of ai models. *Artificial intelligence (AI)*, pp. 4, 2022.

Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. Estimating the carbon footprint of bloom, a 176b parameter language model. *Journal of Machine Learning Research*, 24 (253):1–15, 2023.

Alexandra Sasha Luccioni, Yacine Jernite, and Emma Strubell. Power hungry processing: Watts driving the cost of ai deployment? 2024.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

Jacob Morrison, Clara Na, Jared Fernandez, Tim Dettmers, Emma Strubell, and Jesse Dodge. Holistically evaluating the environmental impact of creating language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=04qx93Viwj.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

Manish Parashar, Tess DeBlanc-Knowles, Erwin Gianchandani, and Lynne E Parker. Strengthening and democratizing artificial intelligence research and development. *Computer*, 56(11):85–90, 2023.

Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. Green ai. *Communications of the ACM*, 63(12):54–63, 2020.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for modern deep learning research. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 13693–13696, 2020.

Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga, Jinshi Huang, Charles Bai, et al. Sustainable ai: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems*, 4:795–813, 2022.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, Chenyang Shao, Yuwei Yan, Qinglong Yang, Yiwen Song, Sijian Ren, Xinyuan Hu, Yu Li, Jie Feng, Chen Gao, and Yong Li. Towards large reasoning models: A survey of reinforced reasoning with large language models, 2025. URL https://arxiv.org/abs/2501.09686.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.