Practical Evaluation of Machine Learning Efficiency Requires Model Life Cycle Assessment

Jared Fernandez **Clara Na**

Yonatan Bisk **Constantine Samaras** **Emma Strubell**

Carnegie Mellon University, Pittsburgh, PA, USA
{jaredfern, clarana, ybisk, csamaras, strubell}@cmu.edu

Abstract

The growing scale of language models entails growing resource requirements and envrionmental impacts. For these systems to have a positive impact on society, it is necessary to thoughtfully weigh the societal and environmental benefits and costs, within the context of a complex model life cycle and many potential measures of impact. In this position paper, we argue the need for **holistic life cycle assessment of language models** across the development and deployment pipeline to properly account for required resources and downstream impact.

1 Introduction

Researchers motivated by empirical scaling laws [62, 39, 31] and beholden to Sutton's "bitter lesson" of AI [71] have achieved breakthroughs in science and technology by leveraging increases in scale of computation across data, model architecture, and hardware platforms [56, 10, 32, 68, 14, 57]. The associated computation has commensurate resource requirements, with multiple projections estimating that data centers will require more than 10% of the total U.S. energy demand by 2030 [26, 7, 67]. Modern models are so large that an individual training run can require 5 million GPU-hours of computation and emit over 1,300 tons of CO_2 equivalent emissions (CO_2e),[1] [2] with even more resources required for experimentation and demand during deployment [52].

The growing complexity of language model development yields significant environmental impact and entails relevant implications for energy security, natural resources, public health, and utility infrastructure [29, 43, 37]. However, the methodologies we use to evaluate machine learning efficiency and its socio-economic impacts have not evolved in kind.

Fortunately, techniques for analyzing the resource requirements and downstream impacts over the lifetime of manufactured products are well established in the field of industrial ecology; namely, with the method of *life cycle assessment (ISO 14040, ISO 14044 [33, 34]).* Life cycle assessment (LCA) quantifies the impact of a product by decomposing its life cycle across the stages of: manufacture, use, and disposal; and across types of resources (e.g. energy, carbon emissions, human health impacts).

Life cycle assessment has been used in semiconductor manufacturing and computing hardware research to quantify the embodied and operational carbon cost of fabrication, recycling, and use of physical hardware[27, 79, 66, 36, 28]. However, systematic methods for application of LCA to machine learning models is nascent. In this position paper, **we enunciate the need for life cycle assessment to evaluate the efficiency and environmental impact of language models through development and deployment**.

[1] https://github.com/meta-llama/llama-models/blob/main/models/llama4/MODEL_CARD.md

[2] Equivalent to the annual CO_2 emissions sequestered by 1,304 acres of forests. Source: USA EPA greehouse gas equivalencies calculator

Socially Responsible Language Modelling Research (SoLaR) Workshop at NeurIPS 2025.

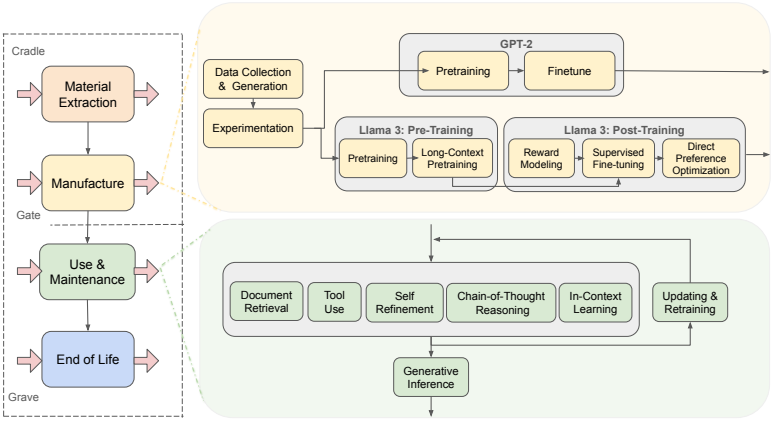


Figure 1: ML model development and deployment life cycles have grown in complexity with increasing numbers of stages. The pre- and post-training pipelines of modern LLMs (e.g. OLMo with the Tulu post training recipe OLMo et al. [55], Lambert et al. [41]) have significantly more stages than classical train-test settings; and a larger variety of methods that can be used for conducting inference [78].

2 Life Cycle Assessment for Language Models

Machine learning model development and deployment pipelines have greatly increased in complexity from classical notions of train and test evaluations. As illustrated in Figure 1, state-of-the-art large language models require multiple stages of pre- and post-training, rely on an assortment of inference-time algorithms, and are deployed across variable software platforms and hardware architectures. Each stage of the growing model pipeline introduces further complexity to decision-making – as well as additional challenges and demands for proper accounting of models' resource consumption and environmental impact.

Previous efforts to account for resource usage and environmental impacts of machine learning models have mainly focused on the singular energy or water use of large-scale model training [69, 59], or the marginal costs of single-example or single-batch inference [69, 48]. Recent investigations have considered the total lifetime energy costs of models during both training and inference [23, 46, 52, 79]; or the costs embodied in their computing hardware [44, 45]. However, there exists variability and uncertainty in the models of interest, training workloads, inference use cases, and deployment settings across these studies; which makes comparisons across works challenging and prone to obsolescence.

Life cycle assessment (LCA; ISO 14040:2006 [33], ISO 14044:2006 [34], Curran [16]) provides a methodological basis for determining the environmental and social impacts of a product by accounting for the required resources and environmental impacts of a manufactured product or service through resource extraction, material processing, manufacture, use, and disposal (i.e. from *cradle to grave*). At the core of LCA is the concept of a *functional unit* which defines a quantitative reference for the value provided by a process, which can be compared across potential systems. Functional units for machine learning models can be defined as appropriate to the focus of study. In turn, a system can be defined that produces the functional unit of interest in relation to resources and emissions.

In this section, we demonstrate how life cycle assessment can be used enable more holistic accounting of machine learning models' total resource consumption and environmental impacts for producing a functional unit. We examine the four stages of LCA as defined in ISO standards: *Goal Definition and Scoping*, *Life Cycle Inventory*, *Life Cycle Impact Assessment*, and *Interpretation.*

2.1 Goal Definition and Scoping

The initial stage of life cycle assessment consists of goal definition and scoping in which functional units are defined and system boundaries drawn; which determines the processes which will be examined and accounted for when calculating the total cost of production for the functional unit.

Life cycle assessment can be used by institutional organizations, machine learning researchers, policy makers, and downstream users to analyze the components of the machine learning model life cycle. Accordingly, the functional unit and the process of interest may vary. For example, institutional developers of large foundation models may be interested in the environmental impact and cost

associated with the development of families of models, and the functional unit could be defined as a "set of trained foundation models for a language task." Downstream users may be concerned with the costs associated with using machine learning models, where a functional unit can be defined as a "processed batch of queries to a machine learning model."

Although prior work has performed direct measures of the operational costs of conducting model training and inference, reported values are not comparable when they are not grounded in standardized functional units. As the functional unit is defined according to its use and performance, it is necessary to ground its specification in the setting in which a model will be used— such as by specifying latency constraints, or required task performance on a benchmark [15, 63, 49, 75, 77, 30]. LCA analyses without well-specified reference flows and functional units may yield misleading conclusions if the analysis fails to account for the operational or embodied costs of model training, or attempts to compare systems with respect to different functional units.

Furthermore, LCA provides two primary approaches for analysis: *attributional* and *consequential*; which provide insight into the total existing environmental impacts of an activity and how environmental impacts will change because of decisions made, respectively [53]. Once the scope of an LCA study is determined with the selection of a functional unit and approach, *system boundaries* can be identified in order to exclude certain stages that are identical across different product systems, and LCA models for the product life cycle can be developed.

Throughout the remainder of this section, we consider an example LCA with a functional unit corresponding to a batch of processed examples by a pretrained large language model.

2.2 Life Cycle Inventory

The life cycle inventory stage describes the environmental flows associated with the functional unit [53]. Examples of inventoried flows include electricity use, consumed water, material flows, and other measures. To estimate the environmental outputs of producing the selected functional unit, *reference flow* diagrams are used to model *product systems* defined over the machine learning model life cycle stages of: material extraction, manufacture, use and maintenance, and disposal. Input resources and output emissions and waste byproducts are associated with each stage. For example, the resources required by client-side resources may be discounted in an LCA comparing different ML models if it can be reasonably assumed that they would be constant regardless of the model.

2.2.1 Reference Flows for Modern Language Model Life Cycles.

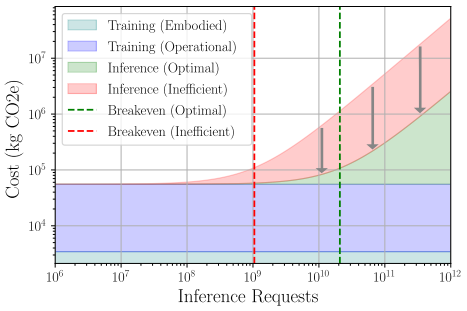
In addition to analyzing the manufacture and disposal of the underlying compute hardware, an LCA analysis of a language model's life cycle requires aggregation and inventory of resource utilization across stages of model development. Historically, the process of developing and deploying models followed a simple process of training and validation on small sets of in-domain i.i.d. datasets.

By contrast, our example functional unit corresponding to modern LLM training and serving requires aggregation of resource flows across more complex training and inference stages; Seen in Figure 1. Modern ML researchers develop models using pipelines with complex experimentation processes and multiple stages of training, such as: automated machine learning and experimentation, pretraining and post-training, continuous retraining and updating of models during deployment [73, 65]. Likewise, the variety of methods for model inference has grown, as new paradigms have emerged which shift computation from training to inference to attain higher performance, e.g. via chain-of-thought reasoning, self-refinement, tool use, retrieval-augmented generation, and in-context learning [78].

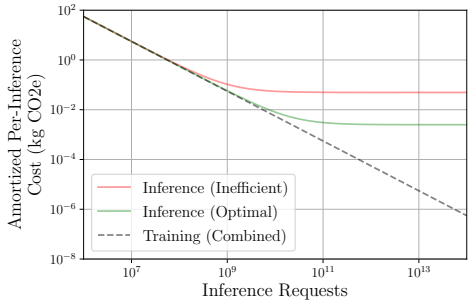
The *life cycle inventory* accounts and attributes resource consumption across constituent stages of model development and deployment. For our example, computing the cost to produce the functional unit C_{FU} requires consideration of not only the marginal cost of inference computation but also the amortized costs of upstream training and hardware manufacturing associated with the inference:

$$C_{\text{FU}} = C_{\text{Per Inference}} + \frac{\text{Hardware Utilization Time} \times C_{\text{Embodied}}}{\text{Hardware Lifespan}} + \frac{C_{\text{Experimentation}} + C_{\text{Training}}}{\text{Total Lifetime Inferences}} \quad (1)$$

Systematic quantification of the resource use in each constituent stage informs comparisons of the relative magnitudes across components of the full model lifecycle. For example, in Figure 2b, we see that the total per-inference cost is extremely sensitive to the total "lifespan" of the model until it is used at least tens of billions of times.



(a) **Aggregate costs:** Total environmental impact of models incorporates factors from all stages of model life cycle.[3]Utilizing efficient serving optimizations increases the number of functional units produced under a fixed resource budget.



(b) **Per-inference costs:** For the functional unit defined as a batch of processed requests, the operational cost of training and embodied costs of hardware are amortized with use and asymptotically approach the marginal cost of inference.

Figure 2: The CO2e emissions of OLMo2 7b training and inference [52, 55]. Increasing inference efficiency via offline batching reduces the unit cost, as does amortization of embodied costs over more model uses. Decomposition of the resource use across life cycle stages enables identification of the *significant issues* (i.e. the life cycle stage which maximally contributes to total costs).

2.2.2 Accounting for Resource Inputs and Output Emissions.

Estimates for the quantity of input raw materials and output by-products and waste consumed at each stage of the model life cycle are needed to quantify the total environmental impact of the machine learning system. With increasing scale of compute resources required for state-of-the-art machine learning models, the energy and resource intensity has increased across all stages of the model life cycle, across hardware fabrication and disposal, and model development and deployment.

The same LCA methodologies can be applied across resource types to account for the costs of a variety of resources and outputs, including raw materials and toxic waste (e.g. PFAS, CFCs) incurred during hardware fabrication, transportation, and disposal [21, 22, 42]. Likewise, LCA estimations can account for the varied operational environmental flows of energy use, water use, and carbon emissions arising from data centers during model experimentation, training, and inference [79, 52, 29].

2.2.3 Comparison of Design Choices and Product Systems

In addition to providing baselines for the resources required by a machine learning model, LCA can be used to evaluate and provide comparisons across multiple systems that produce the same functional unit, and the relative impact of efficiency improvements to stages of the model life cycle.

For modern LLM serving, there exists a variety of design choices that affect the total efficiency and resource consumption, including: parallelization strategies, machine learning software frameworks, cluster scheduling algorithms, and choices in the underlying hardware accelerators. Life cycle assessment enables comparisons of the cost per functional unit when varying such configurations in the context of the full model life time. For instance, shown in Figure 2b, simply increasing the efficiency of LLM serving with increased batching and hardware utilization enables more requests to be served under fixed carbon emissions budgets.

Furthermore, life cycle assessment enables the study of efficiency optimizations that affect multiple stages of the model life cycle. The advent of multi-stage training and inference-time computing yield complex interactions across stages of model development and use which allow for tradeoff of resources between constituent stages. For instance, "reasoning" models (such as DeepSeek-R1, OpenAI GPT-4o, and Gemini) can utilize substantially more resources during inference to attain higher performance on difficult tasks that would otherwise require additional domain-specific training. Alternatively, continual or domain-specific pretraining may extend a model's utility, delaying the need for full model retraining and/or further offsetting the initial training cost. LCA can enable analysis of the trade-offs of these methods, relative efficiencies, and resource-optimal settings.

[3]Following Morrison et al. [52], we ground our inference efficiency estimates in ShareGPT data, and we use the same assumptions as Luccioni et al. [46] to estimate embodied emissions.

2.3 Life Cycle Impact Assessment

Using the quantified costs determined through the life cycle inventory, the total environmental impact of ML models can be determined by translating the inventoried resources into associated impact categories, such as the contribution to global warming from increased emissions; ozone depletion from CFCs; or human health impacts resulting from water depletion, noise, or air quality pollution.

Although LLM developers have begun to report on energy requirements and carbon dioxide emission equivalents (CO2e), the downstream environmental impact remains largely unreported [20, 55]. Fortunately, life cycle impact assessment provides standard conversions and characterization factors for converting inventoried resources into their associated net environmental impact, such as the U.S. Environmental Protection Agency's Tool for the Reduction and Assessment of Chemical and other environmental Impacts (TRACI) [9, 8]. Identifying and quantifying these broader social and environmental impacts is an active area of research in relevant fields of policy, public health, and the study of science, technology and society (STS).

2.4 Interpretation

For researchers, practitioners, and policy makers to utilize the results of an LCA, it is necessary to contextualize and via *interpretation* the results of the investigation, by: (1) identifying significant issues with the inventory and assessment; (2) evaluating the completeness, sensitivity, and consistency of data; and (3) providing conclusions and recommendations based on the impact assessment.

Identification of the *significant issues* (i.e. the components of the life cycle that have the greatest impact on the total result) enables location of resource bottlenecks in machine learning models, whether it be the costs associated with hardware fabrication, the upfront costs associated with model training, or the marginal costs of individual inferences. Once the inventory and impact assessment have been validated, the LCA's results can be used to elect for design choices which reduce the lifetime environmental impact of models — such as to identify which model design choices yield the most efficient system for providing the specified functional unit (e.g., watts per batched inference).

3 Enabling Life Cycle Assessment for Machine Learning

We must provide the necessary specification in our evaluation, transparency in our reporting to meet the requirements of the LCA analysis standards.

User-Centric Evaluations and Metrics Variability in the evaluation settings used to characterize efficiency and performance in ML models hinders fair comparison between studies and models. Moreover, while standard efficiency metrics may be relatively measurable and reproducible (e.g. model parameter count, FLOPs), they often fail to map directly to practical user-side requirements such as latency constraints, financial cost, or energy budget [17, 24, 25]. For functional units to correspond to user needs, efficiency benchmarks should not only measure the hardware utilization or speed but be grounded in the performance measured demanded by the use case.

Transparency in Reporting from Model Developers and Users. As observed in our example in Figure 2b, the cost of a functional unit of inference is directly dependent on: the serving configurations, hardware selection, and ML system design decisions. Likewise, cost of inference is dependent on the total number of inferences served, a necessary datapoint needed for appropriate attribution of resulting implications to the amortizable training and embodied costs.

While it is increasingly common practice for developers to release information on total energy use and estimated carbon emission equivalents for model pretraining, such measurements are often limited to the final training run and fail to account for development costs, the operational impacts of hardware, or the cost and frequency of inference. For downstream users and regulators to accurately assess the cost of ML models, it is necessary for hardware manufacturers and large-scale model developers to release information on the *embodied resources and emissions* associated with hardware fabrication; as well as the *scale, frequency, and settings for model inference*.

Fortunately, there is precedent for transparency by large-scale industry institutions with the advent of the Foundation Model Transparency Index and model cards [11, 51]. We advocate for a voluntary inclusion of these crucial ingredients for model LCA in model cards whenever possible. Notably,

as full inference details are typically indeterminable before model deployment, we call for users deploying models to periodically release updated information about their models' downstream usage.

4 Implications and Benefits of LCA for LMs and ML

LCA Empowers Decision Making and Effective Resource Allocation. With the growing scale of model training and frequency of inference, further growth in machine learning is becoming constrained by fundamental limitations in the availability of computing hardware and the energy necessary to power them. Life cycle assessment provides insight into the relative resource consumption and intensity of model training and deployment. By identifying *significant issues* (see §2.4), LCA can draw attention to research questions and directions addressing elements of the model life cycle that present the largest bottlenecks to efficiency and opportunities for improvement.

Additionally, LCA enables industry stakeholders to provision and allocate resources so that machine learning systems meet target efficiency and environmental goals — not just in terms of the marginal cost of training or inference, but contextualized within the whole machine learning life cycle. Understanding the relative scales of demand for different life cycle stages enables infrastructure developers to project and accommodate the resources requirements of different workloads (e.g. adapting compute and electrical infrastructure to handle synchronous training or online inference).

LCA Improves Accuracy and Completeness in Resource Estimation and Projections. While the speed and computing requirements of machine learning research have grown with time, the methods required to evaluate the efficiency and resource consumption of the work have not kept up. It is necessary to develop a methodological foundation for grounded assessments of cost.

As shown in Figure 2b, LCA can be used to allocate resource usage to components, and to estimate the relative importance of the constituent stages of hardware fabrication, model training and inference. Additionally, by applying LCA across different types of resources, researchers can account for machine learning's environmental burden along with other key impacts commonly associated with costs of computing such as raw material extraction [12], water usage[43], public health [29], and per- or polyfluoroalkyl substances (PFAS) [42, 21].

Furthermore, LCA enables researchers to estimate future machine learning systems and provides a tool to understand their potential environmental impact, longterm trends, and rebound effects across a range of scenarios [47]. Evaluating hypothetical systems with differing assumptions enables projection of the impact of: further scaling of ML systems [31, 62], automation of the development process with autoML [73], or alternative hardware platforms such as in edge or mobile settings [61].

LCA Informs Environmental Impact Accounting for Policy Makers and the General Public. As the use of AI has grown and energy, water, and other impacts have materialized in many communities, policy makers at federal and local levels are increasingly interested in assessing and mitigating impacts. The energy, carbon, water, air pollutant, noise, and other impacts of AI data centers has driven interest from policy makers and communities for solutions. Lawmakers have introduced bills to evaluate the environmental impacts of AI and establish standardized reporting systems e.g. [3]. Yet the environmental impacts, and potential benefits, of AI extend beyond the direct impact of computing and data centers, and also how AI and ML use affect applications in infrastructure, as well as extend to broader societal systems [38]. Both attributional and consequential LCA provide useful methods for policy makers and communities to estimate and interpret the broad contours of how AI and ML affect the environment and society and set guidelines for improved outcomes.

Although there is no standardized method to conduct an LCA, transparently defining scope and functional units can enable guidelines for voluntary or regulated impacts reporting from industry stakeholders, and inform policy maker decisions. For example, the U.S. Inflation Reduction Act [1] specifies the use of LCA and a model developed by Argonne National Laboratory as the method required to estimate the life cycle greenhouse emissions of hydrogen production to determine eligibility for federal incentives. As policymakers consider both incentives and regulations to minimize the environmental impacts of AI and ML, LCA can enable model comparison and account for both impact and performance. These incentives and regulations can also inform industry decision-making regarding model development to account for resource requirements and external impact.

References

[1] Inflation reduction act of 2022. Public Law 117-167, 117th Congress, 2022.

[2] *HotCarbon '23: Proceedings of the 2nd Workshop on Sustainable Computer Systems*, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702426.

[3] Artificial intelligence environmental impacts act of 2024. S.3732, 118th Congress, 2024.

[4] *Fourth Workshop on Efficient Natural Language and Speech Processing*, Vancouver, Canada, 2024.

[5] *Workshop on Efficient Systems for Foundation Models II*, Vienna, Austria, 2024.

[6] *First Workshop on Green Foundation Models*, Milan, Italy, 2024.

[7] J. Aljbour, T. Wilson, and P. Patel. Powering intelligence: Analyzing artificial intelligence and data center energy consumption. *EPRI White Paper no. 3002028905*, 2024.

[8] J. Bare. Traci 2.0: the tool for the reduction and assessment of chemical and other environmental impacts 2.0. *Clean Technologies and Environmental Policy*, 13:687–696, 2011.

[9] J. C. Bare. Traci: The tool for the reduction and assessment of chemical and other environmental impacts. *Journal of industrial ecology*, 6(3-4):49–78, 2002.

[10] M. Bobrowsky. Meta spending to soar on ai, massive data center. *Wall Street Journal*, 2025.

[11] R. Bommasani, K. Klyman, S. Longpre, S. Kapoor, N. Maslej, B. Xiong, D. Zhang, and P. Liang. The foundation model transparency index. *arXiv preprint arXiv:2310.12941*, 2023.

[12] S. B. Boyd. *Life-cycle assessment of semiconductors*. Springer Science & Business Media, 2011.

[13] H. J. Byun, U. Gupta, and J.-S. Seo. Energy-/carbon-aware evaluation and optimization of 3d ic architecture with digital compute-in-memory designs. *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, 2024.

[14] K. Cai and D. M. Sophia. Alphabet plans massive capex hike, reports cloud revenue growth slowed. *Reuters*, 2025.

[15] J. Cui, Z. Li, L. Xing, and X. Liao. Safeguard-by-development: A privacy-enhanced development paradigm for multi-agent collaboration systems. *arXiv preprint arXiv:2505.04799*, 2025.

[16] M. A. Curran. *Life-cycle assessment: principles and practice*. National Risk Management Research Laboratory, Office of Research and Development, U.S. Environmental Protection Agency, 2006.

[17] M. Dehghani, Y. Tay, A. Arnab, L. Beyer, and A. Vaswani. The efficiency misnomer. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=iulEMLYh1uR`.

[18] M. Dehghani, Y. Tay, A. Arnab, L. Beyer, and A. Vaswani. The efficiency misnomer. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=iulEMLYh1uR`.

[19] Y. Ding and T. Shi. Sustainable llm serving: Environmental implications, challenges, and opportunities. In *2024 IEEE 15th International Green and Sustainable Computing Conference (IGSC)*, pages 37–38. IEEE Computer Society, 2024.

[20] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[21] M. Elgamal, D. Carmean, E. Ansari, O. Zed, R. Peri, S. Manne, U. Gupta, G.-Y. Wei, D. Brooks, G. Hills, et al. Cordoba: Carbon-efficient optimization framework for computing systems. In *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 1289–1303. IEEE, 2025.

[22] M. Elgamal, A. Mahmoud, G.-Y. Wei, D. Brooks, and G. Hills. Modeling pfas in semiconductor manufacturing to quantify trade-offs in energy efficiency and environmental impact of computing systems. *arXiv preprint arXiv:2505.06727*, 2025.

[23] A. Faiz, S. Kaneda, R. Wang, R. Osi, P. Sharma, F. Chen, and L. Jiang. LLMCarbon: Modeling the end-to-end Carbon Footprint of Large Language Models, Jan. 2024. URL `http://arxiv.org/abs/2309.14393`. arXiv:2309.14393 [cs].

[24] J. Fernandez, J. Kahn, C. Na, Y. Bisk, and E. Strubell. The framework tax: Disparities between inference efficiency in nlp research and deployment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1588–1600, 2023.

[25] J. Fernandez, C. Na, V. Tiwari, Y. Bisk, S. Luccioni, and E. Strubell. Energy considerations of large language model inference and efficiency optimizations. In *Submitted to Association for Computational Linguistics Rolling Review*, 2025.

[26] A. Green, H. Tai, J. Noffsinger, and P. Sachdeva. How data centers and the energy sector can sate ai's hunger for power. *McKinsey and Company*, 2024.

[27] U. Gupta, Y. G. Kim, S. Lee, J. Tse, H.-H. S. Lee, G.-Y. Wei, D. Brooks, and C.-J. Wu. Chasing carbon: The elusive environmental footprint of computing. In *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 854–867. IEEE, 2021.

[28] U. Gupta, M. Elgamal, G. Hills, G.-Y. Wei, H.-H. S. Lee, D. Brooks, and C.-J. Wu. Act: Designing sustainable computer systems with an architectural carbon modeling tool. In *Proceedings of the 49th Annual International Symposium on Computer Architecture*, pages 784–799, 2022.

[29] Y. Han, Z. Wu, P. Li, A. Wierman, and S. Ren. The unpaid toll: Quantifying the public health impact of ai. *arXiv preprint arXiv:2412.06288*, 2024.

[30] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[31] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, et al. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 30016–30030, 2022.

[32] M. Isaac. Meta to increase spending to $65 billion this year in a.i. push. *New York Times*, 2025.

[33] ISO 14040:2006. Environmental management – Life cycle assessment – Principles and framework, 2006.

[34] ISO 14044:2006. Environmental management – Life cycle assessment – Requirements and guidelines, 2006.

[35] W. S. Jevons. The coal question. In *The Economics of Population*, pages 193–204. Routledge, 1866.

[36] S. Ji, Z. Yang, X. Chen, S. Cahoon, J. Hu, Y. Shi, A. K. Jones, and P. Zhou. Scarif: Towards carbon modeling of cloud servers with accelerators. In *2024 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 496–501. IEEE, 2024.

[37] Joint Legislative Audit and Review Commission. Data Centers in Virginia. Technical Report 598, Commonwealth of Virginia, 2024.

[38] L. H. Kaack, P. L. Donti, E. Strubell, G. Kamiya, F. Creutzig, and D. Rolnick. Aligning artificial intelligence with climate change mitigation. *Nature Climate Change*, 12(6):518–527, 2022.

[39] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[40] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pages 5637–5664. PMLR, 2021.

[41] N. Lambert, J. Morrison, V. Pyatkin, S. Huang, H. Ivison, F. Brahman, L. J. V. Miranda, A. Liu, N. Dziri, S. Lyu, Y. Gu, S. Malik, V. Graf, J. D. Hwang, J. Yang, R. L. Bras, O. Tafjord, C. Wilhelm, L. Soldaini, N. A. Smith, Y. Wang, P. Dasigi, and H. Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training, 2025. URL https://arxiv.org/abs/2411.15124.

[42] J. C. Lee, S. Smaoui, J. Duffill, B. Marandi, and T. Varzakas. Forever chemicals pfas global impact and activities, cascading consequences of colossal systems failure: Long-term health effects, food-systems, eco-systems. 2025.

[43] P. Li, J. Yang, M. A. Islam, and S. Ren. Making AI Less "Thirsty": Uncovering and Addressing the Secret Water Footprint of AI Models, Jan. 2025. URL http://arxiv.org/abs/2304.03271. arXiv:2304.03271 [cs].

[44] Y. Li, Z. Hu, E. Choukse, R. Fonseca, G. E. Suh, and U. Gupta. Ecoserve: Designing carbon-aware ai inference systems. *arXiv preprint arXiv:2502.05043*, 2025.

[45] Y. L. Li, O. Graif, and U. Gupta. Towards carbon-efficient llm life cycle. In *Proceedings of the 3rd Workshop on Sustainable Computer Systems (HotCarbon)*, 2024.

[46] A. S. Luccioni, S. Viguier, and A.-L. Ligozat. Estimating the carbon footprint of bloom, a 176b parameter language model. *Journal of Machine Learning Research*, 24(253):1–15, 2023.

[47] A. S. Luccioni, E. Strubell, and K. Crawford. From Efficiency Gains to Rebound Effects: The Problem of Jevons' Paradox in AI's Polarized Environmental Debate, Jan. 2025. URL http://arxiv.org/abs/2501.16548. arXiv:2501.16548 [cs].

[48] S. Luccioni, Y. Jernite, and E. Strubell. Power hungry processing: Watts driving the cost of ai deployment? In *Proceedings of the 2024 ACM conference on fairness, accountability, and transparency*, pages 85–99, 2024.

[49] P. Mattson, C. Cheng, G. Diamos, C. Coleman, P. Micikevicius, D. Patterson, H. Tang, G.-Y. Wei, P. Bailis, V. Bittorf, et al. Mlperf training benchmark. *Proceedings of Machine Learning and Systems*, 2:336–349, 2020.

[50] G. Menghani. Efficient deep learning: A survey on making deep learning models smaller, faster, and better. *ACM Comput. Surv.*, 55(12), Mar. 2023. ISSN 0360-0300. doi: 10.1145/3578938. URL https://doi.org/10.1145/3578938.

[51] M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, and T. Gebru. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 220–229, 2019.

[52] J. Morrison, C. Na, J. Fernandez, T. Dettmers, E. Strubell, and J. 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.

[53] National Academies of Sciences, Engineering, and Medicine and others. *Current Methods for Life-Cycle Analyses of Low-Carbon Transportation Fuels in the United States*. 2022.

[54] S. Nguyen, B. Zhou, Y. Ding, and S. Liu. Towards sustainable large language model serving. *ACM SIGENERGY Energy Informatics Review*, 4(5):134–140, 2024.

[55] T. OLMo, P. Walsh, L. Soldaini, D. Groeneveld, K. Lo, S. Arora, A. Bhagia, Y. Gu, S. Huang, M. Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

[56] OpenAI. Announcing the stargate project, 2025.

[57] M. Parashar, T. DeBlanc-Knowles, E. Gianchandani, and L. E. Parker. Strengthening and democratizing artificial intelligence research and development. *Computer*, 56(11):85–90, 2023.

[58] P. Patel, E. Choukse, C. Zhang, Í. Goiri, B. Warrier, N. Mahalingam, and R. Bianchini. Characterizing power management opportunities for llms in the cloud. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, pages 207–222, 2024.

[59] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021.

[60] D. Patterson, J. Gonzalez, U. Hölzle, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. R. So, M. Texier, and J. Dean. The carbon footprint of machine learning training will plateau, then shrink. *Computer*, 55(7):18–28, 2022. URL `https://arxiv.org/abs/2204.05149`.

[61] D. Patterson, J. M. Gilbert, M. Gruteser, E. Robles, K. Sekar, Y. Wei, and T. Zhu. Energy and emissions of machine learning on smartphones vs. the cloud. *Communications of the ACM*, 67 (2):86–97, 2024.

[62] J. W. Rae, S. Borgeaud, T. Cai, K. Millican, J. Hoffmann, F. Song, J. Aslanides, S. Henderson, R. Ring, S. Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

[63] V. J. Reddi, C. Cheng, D. Kanter, P. Mattson, G. Schmuelling, C.-J. Wu, B. Anderson, M. Breughe, M. Charlebois, W. Chou, et al. Mlperf inference benchmark. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, pages 446–459. IEEE, 2020.

[64] N. Sadat Moosavi, I. Gurevych, Y. Hou, G. Kim, Y. J. Kim, T. Schuster, and A. Agrawal, editors. *Proceedings of the Fourth Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*, Toronto, Canada (Hybrid), July 2023. Association for Computational Linguistics. URL `https://aclanthology.org/2023.sustainlp-1.0/`.

[65] V. Sangarya, R. Bradford, and J.-E. Kim. Estimating environmental cost throughout model's adaptive life cycle. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, volume 7, pages 1281–1291, 2024.

[66] I. Schneider, H. Xu, S. Benecke, D. Patterson, K. Huang, P. Ranganathan, and C. Elsworth. Life-cycle emissions of ai hardware: A cradle-to-grave approach and generational trends. *arXiv preprint arXiv:2502.01671*, 2025.

[67] A. Shehabi, A. Hubbard, A. Newkirk, N. Lei, M. A. B. Siddik, B. Holecek, J. Koomey, E. Masanet, D. Sartor, et al. 2024 united states data center energy usage report, 2024.

[68] B. Smith. The golden opportunity for american ai, 2025.

[69] E. Strubell, A. Ganesh, and A. McCallum. Energy and policy considerations for modern deep learning research. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13693–13696, 2020.

[70] Y. Sui, Y.-N. Chuang, G. Wang, J. Zhang, T. Zhang, J. Yuan, H. Liu, A. Wen, S. Zhong, H. Chen, and X. Hu. Stop overthinking: A survey on efficient reasoning for large language models, 2025. URL `https://arxiv.org/abs/2503.16419`.

[71] R. Sutton. The bitter lesson, 2019.

[72] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler. Efficient transformers: A survey. *ACM Comput. Surv.*, 55(6), Dec. 2022. ISSN 0360-0300. doi: 10.1145/3530811. URL `https://doi.org/10.1145/3530811`.

[73] T. Tornede, A. Tornede, J. Hanselle, F. Mohr, M. Wever, and E. Hüllermeier. Towards green automated machine learning: Status quo and future directions. *Journal of Artificial Intelligence Research*, 77:427–457, 2023.

[74] M. Treviso, J.-U. Lee, T. Ji, B. van Aken, Q. Cao, M. R. Ciosici, M. Hassid, K. Heafield, S. Hooker, C. Raffel, P. H. Martins, A. F. T. Martins, J. Z. Forde, P. Milder, E. Simpson, N. Slonim, J. Dodge, E. Strubell, N. Balasubramanian, L. Derczynski, I. Gurevych, and R. Schwartz. Efficient methods for natural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 11:826–860, 2023. doi: 10.1162/tacl_a_00577. URL `https://aclanthology.org/2023.tacl-1.48/`.

[75] A. Tschand, A. T. R. Rajan, S. Idgunji, A. Ghosh, J. Holleman, C. Kiraly, P. Ambalkar, R. Borkar, R. Chukka, T. Cockrell, et al. Mlperf power: Benchmarking the energy efficiency of machine learning systems from microwatts to megawatts for sustainable ai. *arXiv preprint arXiv:2410.12032*, 2024.

[76] Z. Wan, X. Wang, C. Liu, S. Alam, Y. Zheng, et al. Efficient large language models: A survey. *arXiv preprint arXiv:2312.03863*, 1, 2023.

[77] Y. Wang, X. Ma, G. Zhang, Y. Ni, A. Chandra, S. Guo, W. Ren, A. Arulraj, X. He, Z. Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[78] S. Welleck, A. Bertsch, M. Finlayson, H. Schoelkopf, A. Xie, G. Neubig, I. Kulikov, and Z. Harchaoui. From decoding to meta-generation: Inference-time algorithms for large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL `https://openreview.net/forum?id=eskQMcIbMS`. Survey Certification.

[79] C.-J. Wu, R. Raghavendra, U. Gupta, B. Acun, N. Ardalani, K. Maeng, G. Chang, F. Aga, J. Huang, C. Bai, et al. Sustainable ai: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems*, 4:795–813, 2022.

[80] Y. Wu, I. Hua, and Y. Ding. Unveiling environmental impacts of large language model serving: A functional unit view. *arXiv preprint arXiv:2502.11256*, 2025.

[81] H. Yao, C. Choi, B. Cao, Y. Lee, P. W. W. Koh, and C. Finn. Wild-time: A benchmark of in-the-wild distribution shift over time. *Advances in Neural Information Processing Systems*, 35:10309–10324, 2022.

[82] J. Zheng, X. Cai, Q. Li, D. Zhang, Z. Li, Y. Zhang, L. Song, and Q. Ma. Lifelongagentbench: Evaluating llm agents as lifelong learners. *arXiv preprint arXiv:2505.11942*, 2025.

A Broader Impacts & Limitations

Throughout our work, we utilize different large language model development and deployment flows which we note are examples of current machine learning workloads but are not representative of all development and deployment pipelines. Complexity of pipelines will evolve over time, model "manufacture" and "use" stages will differ across architectures. Other considerations beyond energy and carbon are associated; we direct the machine learning research community towards LCA as a generalizable methodology which can be applied across different forms of resources.

Furthermore, it is necessary to make reductive assumptions as to the nature of the resources consumed during any assessment – which may neglect other components in the ML model life cycle; such as how the power demands may affect the power providing infrastructure and the electrical grid.

B Status Quo of Machine Learning Efficiency and its Limitations

In response to recent growth in ML's computational demands and concerns around its corresponding resource consumption, there has been a significant increase in scientific inquiry towards efficient ML methods in recent years; as reflected in a myriad of research surveys [50, 74, 72, 76, 70] and publication venues dedicated to the topic [2, 4, 5, 6, 64]. The status quo of quantifying ML efficiency is well established, and represents alternative views to the position that we posit in this paper. We describe this status quo, and its limitations, in more detail in the following section.

B.1 Challenges in Measuring Model Efficiency

Growing Complexity and Variability of the Model Life Cycle. With the growing scale and associated computational demands of foundation models, recent efforts have sought to characterize stages of model development training [60, 69], and deployment inference [58, 80, 19, 54]. Such efforts have led institutional model developers to release estimates of the associated energy costs of model development (e.g. Meta Llama-3 and the AI2's OLMo models [20, 52]). However, focus on individual stages of the model life cycle is insufficient to measure the total resources and environmental impact associated with the choice to build a new machine learning model or AI system.

Recent works have estimated the power utilization across the model life cycle stages of both training and inference [79, 46]. Unfortunately, a lack of standardization of the accounted stages makes comparison across studies difficult. Furthermore, complexity of the model development and deployment use cases has grown beyond classical settings of train-test evaluations; with models now requiring multiple stages of pretraining and post-training, as well as variability in inference-time compute with the advent of "reasoning models". A standardized framework is needed with explicit definitions of the stages of deployment and use in order to enable accurate reporting.

Insufficient Proxy Measures of Machine Learning Efficiency. A wide array of efficiency metrics have motivated research in the design of efficient machine learning algorithms, model architectures, and computer systems. For example, service-level objectives (SLOs) have been used to optimize cloud serving settings where models are deployed to support latency-sensitive APIs. Whereas the hardware limitations of mobile and edge settings have yielded model compression methods which reduce the memory overheads of models. At the same time, theoretical investigations, which are often based on proxy metrics for efficiency such as FLOPs, have yielded parameter-, data-, and sample-efficient ML architectures and training algorithms. Although such research demonstrates improvements towards targeted objectives, proxy measures of efficiency are often not highly correlated with more tangible measures such as latency and energy [18, 24].

Furthermore, existing models are often developed to optimize performance on static benchmarks. In reality, models are deployed in continuously changing environments which require repeated retraining or continual updating. However, existing efficiency evaluations of robustness to distribution shift fail to account for the efficiency and resources needed to continuously update models [81, 40, 82].

B.2 Distinction Between Model and Hardware Life Cycles.

Life cycle assessment of the computing hardware utilized in machine learning [27, 66] provides valuable information on the embodied emissions and operational carbon of the hardware platforms. Minimization of carbon emissions has been used to optimize the design of efficient computing hardware and architectures [28, 21, 13]. Analysis of the environmental costs associated with hardware provides insight into the efficiency of the underlying computing platform and informs decision-making around hardware provisioning and infrastructure design decisions.

Although the design of energy-efficient and carbon-efficient computing architectures reduces the lifetime environmental impact of computing hardware, hardware-based accounting alone does not provide insight into the resource requirements and associated emissions of the *machine learning model* which is often developed and deployed across multiple heterogeneous hardware platforms over the course of its lifetime.

B.3 Disconnects Between Individual Computational Workloads and Sector-Wide Projections.

Sector-Level Estimates are Too Coarse-Grained for Estimating the Impact of Individual Model Life Cycles. Concerns around the rising power demands of AI data centers have led to the rise of various studies that estimate and project growth in data center energy use [67, 26, 7]. To obtain projections on energy use, such studies rely on estimates of future chip shipments and energy efficiency to forecast the total demands of computing hardware. Sector-level analysis is critical for providing information to developers of electrical grid infrastructure. With infrastructure lead times of multiple years, accurate sector projections enables grid infrastructure to be built out to support the increased capacity demands of data centers, often in excess.

However, these studies rely on assumptions about hardware utilization and energy efficiency at a level of abstraction that obfuscates individual workloads. These assumptions make it impossible to assess for the impact of models developed and deployed by machine learning researchers and practitioners; or to evaluate the impact of model efficiency improvements or design choices.

Efficiency Improvements Will Not Keep Energy Use Increases in Check. Although algorithmic efficiency and per-accelerator energy efficiency (i.e. FLOPS per watt) have increased over time, so has the embodied carbon of the GPU accelerators [60, 44]. Furthermore, the increased performance and efficiency of AI systems in real-world use cases yields *rebound effects* such as Jevons' paradox [35] in which there is increased uptake, leading to increased total resource consumption despite higher utilization and efficiency and reductions in resources consumed per-unit of resource [47].

With the large number of factors governing the efficiency of systems, ranging from hardware to software to algorithmic efficiency, improvements in efficiency with respect to a single one of these components alone cannot be relied upon to mitigate the total costs of machine learning. Under the assumption of Jevons' paradox and increased capability and profitability,[4] ML demand will expand to consume all resources that can be allocated to it; under this setting, managing ML's resource consumption becomes less a question of reducing resource use, than in resource allocation: Given a limited set of resources (e.g. datacenter energy, land), what is the most effective allocation of those resources in order to maximize output? There is a need for methodologies and data enabling analysis of such resource allocations, e.g. between training and inference workloads, across different model types (task-specific, general-purpose), tasks, deployment scenarios, and hardware.

[4] As of May 2025 Google cites a 50x increase in use of their AI platform year-over-year since 2024 (9.7T to over 480T tokens processed monthly), and Microsoft's CEO recently referenced Jevons' paradox to reassure stakeholders that ML efficiency improvements will lead to "skyrocket[ing]" demand for AI, not to mention significant investments in energy infrastructure, such as Microsoft's bid to re-commission the nuclear reactor at Three Mile Island in order to power an AI data center.