AI Benchmark Democratization and Carpentry

Gregor von Laszewski,[1] Wesley Brewer,[2] Jeyan Thiyagalingam,[3] Juri Papay,[3] Armstrong Foundjem,[4] Piotr Luszczek,[5] Murali Emani,[6] Shirley V. Moore,[7] Vijay Janapa Reddi,[8] Matthew D. Sinclair,[9] Sebastian Lobentanzer,[10] Sujata Goswami,[11] Benjamin Hawks,[12] Marco Colombo,[13] Nhan Tran,[12] Christine R. Kirkpatrick,[14] Abdulkareem Alsudais,[15] Gregg Barrett,[16] Tianhao Li,[17] Kirsten Morehouse,[18] Shivaram Venkataraman,[9] Rutwik Jain,[9] Kartik Mathur,[20] Victor Lu,[21] Tejinder Singh,[22] Khojasteh Z. Mirza,[23] Kongtao Chen,[24] Sasidhar Kunapuli,[25] Gavin Farrell,[26] Renato Umeton,[27] Geoffrey C. Fox[1]

[1] *Biocomplexity Institute, University of Virginia, Charlottesville, VA, USA*
[2] *Oak Ridge National Laboratory, Oak Ridge, TN, USA*
[3] *Rutherford Appleton Laboratory, STFC, Harwell Campus, UK*
[4] *DEEL, Polytechnique Montreal, Montreal, Canada*
[5] *LLSC, MIT Lincoln Laboratory, Lexington, MA, USA*
[6] *Argonne National Laboratory, Lemont, IL, USA*
[7] *Computer Science Department, UTEP, El Paso, TX, USA*
[8] *Harvard University, Boston, MA, USA*
[9] *Computer Sciences Department, Univ. of Wisconsin–Madison, Madison, WI, USA*
[10] *Helmholtz Center Munich, Munich, Germany*
[11] *ALS, LBNL, Berkeley, CA, USA*
[12] *Fermilab, Batavia, IL, USA*
[13] *Discovery Partners Institute, UIUC, Chicago, IL, USA*
[14] *SDSC, UC San Diego, San Diego, CA, USA*
[15] *Prince Sattam bin Abdulaziz University, Saudi Arabia*
[16] *Cirrus AI, Johannesburg, South Africa*
[17] *Duke University, Durham, NC, USA*
[18] *Harvard University, Cambridge, MA, USA*
[20] *Microsoft, Vancouver, BC, Canada*
[21] *Independent Researcher, Tampa, FL, USA*
[22] *Office of the CTO, Dell Technologies, Santa Clara, CA, USA*
[23] *Cornell Tech, Cornell University, New York, NY, USA*
[24] *Google, Mountain View, CA, USA*
[25] *Independent Researcher, San Jose, CA, USA*
[26] *University of Padua, Padua, Italy*
[27] *St. Jude Children's Research Hospital, Memphis, TN*

Correspondence*:
Gregor von Laszewski
laszewski@gmail.com

ABSTRACT

Benchmarks are one cornerstone of modern machine learning practice, providing standardized evaluations that enable reproducibility, comparison, and scientific progress. However, AI benchmarks are becoming increasingly complex, requiring special care, including AI focused dynamic workflows. This is evident by the rapid evolution of AI models in architecture, scale, and capability; the evolution of datasets; and deployment contexts continuously change, creating a moving target for evaluation. Large language models in particular are known for their memorization of static benchmarks, which causes a drastic difference between benchmark results and real-world performance. Beyond the accepted static benchmarks we know from the traditional computing community, we need to develop and evolve continuous adaptive benchmarking frameworks, as scientific assessment is increasingly misaligned with real-world deployment risks. This requires the development of skills and education focused on benchmarks in the scientific community: *AI Benchmark Carpentry*.

Drawing on our experience from MLCommons, educational initiatives, and government programs such as the DOE's Trillion Parameter Consortium, we identify key barriers that hinder the broader adoption, utility, and evolution of benchmarking in AI. These include substantial resource demands, limited access to specialized hardware, lack of expertise in benchmark design, and uncertainty among practitioners about how to relate benchmark results to their own application domains. Moreover, current benchmarks often emphasize peak performance on leadership-class hardware, offering limited guidance for more diverse, real-world deployment scenarios. This may include applications to smaller compute resources, but also to larger systems such as LLMs deployed by commercial entities.

We argue that benchmarking itself must become dynamic in order to incorporate evolving models, updated data, and heterogeneous computational platforms while maintaining transparency, reproducibility, and interpretability. Democratizing this process requires not only technical innovation, but also systematic educational efforts as part of AI benchmark carpentry offerings, spanning undergraduate to professional levels, in order to develop sustained expertise in benchmark design and use. Finally, benchmarks should be framed and used to support application-relevant comparisons, enabling both developers and users to make informed, context-sensitive decisions. Advancing dynamic and inclusive benchmarking practices will be essential to ensure that evaluation keeps pace with the evolving AI landscape and supports responsible, reproducible, and accessible AI deployment. Furthermore, we believe that it is timely to provide a solid foundation for designing, using, and evolving benchmarks through community efforts that allows us to enable the concept of *AI benchmark carpentry.*

Keywords: benchmark, AI benchmark, AI benchmark carpentry, AI benchmark democratization, MLCommons

1 INTRODUCTION

Recently, the availability of graphics processing units (GPUs) and the rapid progress in artificial intelligence (AI) – especially in the area of deep learning – have brought a revolution to the scientific community. However, the use of these technologies is still in its infancy due to several factors. First, many application scientists are unsure how to leverage these newly available tools and instruments. Second, it remains unclear what level of effort is required to integrate them into their own research. Third, the specific demands these technologies place on infrastructure to be useful for a given scientific problem are not yet well understood.

Some of these challenges can be addressed by providing meaningful benchmarks to the scientific community, which can help researchers assess the usefulness and scalability of AI methods for their own applications. Therefore, it is beneficial to formalize the development of standardized AI benchmarks—not by a few individuals, but by the broader community. Such benchmarks can serve as a critical foundation for the scientific community, enabling rigorous evaluation, comparison, and reproducibility of new models and techniques.

However, as AI systems have become more sophisticated, incorporating complex and dynamic workflows, the traditional static approach to defining benchmarks has proven to be a significant limitation. In addition, to conventional benchmarks that capture key concepts familiar to scientists, we must also account for the continuous evolution of AI models and architectures, the changing nature of datasets, and the diversity of deployment contexts. These factors create a moving target for evaluation, risking a growing misalignment between benchmark results and the actual performance of AI systems in real-world scenarios.

Drawing on insights from our work with MLCommons, educational initiatives, and government-led projects such as the U.S. Department of Energy's Trillion Parameter Consortium [1, 2], we identify a set of fundamental barriers that impede the broader utility and adoption of AI benchmarking. Beyond the substantial resource demands and limited access to specialized, leadership-class hardware, there exists a pervasive lack of expertise in benchmark design and a growing uncertainty among practitioners regarding how to relate these performance metrics to their specific application domains. Current benchmarks—by often prioritizing peak performance on elite hardware—offer insufficient guidance for the diverse range of computational platforms encountered in practice, from smaller-scale devices to large, pre-deployed commercial language models.

This paper argues that the practice of AI benchmarking itself must become dynamic and adaptable to keep pace with the rapidly evolving AI landscape. To achieve this, benchmarks must be designed to transparently incorporate evolving models, updated datasets, and heterogeneous computational platforms, while upholding the core principles of transparency, reproducibility, and interoperability. We propose that two complementary strategies can advance this goal: first, democratizing the creation of AI benchmarks and expanding the community contributing to them; and second, establishing a robust foundation for the technical execution and innovation of benchmarks through coordinated educational efforts. Together, these approaches will foster sustained expertise spanning from undergraduate education to professional practice.

We believe it is both timely and necessary to establish a solid foundation for the design, use, and evolution of benchmarks through collaborative community efforts—thereby enabling what we call AI benchmark carpentry. This paper summarizes the collective perspectives developed through this process within the MLCommons Science & HPC Working Group.

The paper is organized as follows. In Section 2, we introduce some essential definitions that we use throughout this paper. Section 3 introduces a formal specification for AI benchmarks. In Section 4, we summarize briefly some existing AI benchmark efforts. In Section 5, we outline how to share benchmarks. In Section 6, we define activities to be conducted as part of the educational efforts. In Section 7, we identify what we need to do to conduct democratization efforts. Lastly, we conclude in Section 8.

Additionally, we list acronyms and abbreviations used in this paper in the Appendix A. Contributions of the authors are summarized in the Appendix B.

2 DEFINITIONS

In this section, we introduce some of the definitions and terminology used throughout this work in order to work towards a formal definition of AI benchmarks.

2.1 What is Benchmarking?

In computing and scientific software evaluation, benchmarking is the process of comparing metrics for computer programs, models, or systems in order to assess their relative performance, typically with respect to a baseline. While early benchmarks focused largely on hardware throughput (e.g., the time required to complete a fixed computational task), modern benchmarks increasingly evaluate software, algorithms, and integrated systems. Three dimensions now structure most benchmarking efforts: 1) runtime—the amount of time a system requires to complete a set task; 2) accuracy—the comparative quality or correctness of outcomes for the same task; and 3) efficiency—the ratio between used computational resources and quality of outcomes.

The goals of benchmarking include identifying performance gaps, establishing baseline expectations, driving innovation, and supporting continuous improvement over both short- and long-term horizons. Benchmarking has been extensively used in computer engineering and science—across both industry and academia—to measure the performance of computing equipment and the applications running on such systems.

In addition to the classical primary outcome metrics (runtime, accuracy, efficiency), today's benchmarks evaluate secondary qualities that are of high importance to the real-world deployment of systems. These include robustness and reliability (stability with respect to distribution shifts and noise, generalization), usability and accessibility (ease of integration with other systems, error transparency, ease of setup), and reproducibility (stability of the results and consistent behavior across versions, seeds, or environments).

2.2 Lessons Learned from Traditional HPC Benchmarking

Traditional high-performance computing (HPC) benchmarking includes:

1. *synthetic benchmarks* that simulate characteristic community workloads, as exemplified by the TOP500 and Green500 benchmarks;
2. *application benchmarks* that represent real-world applications to measure end-to-end performance, such as SPEC HPC; and
3. *scientific application benchmarks* that emphasize the accuracy of computational methods in solving domain-specific scientific problems.

(For a more detailed discussion, see Section 4.1)

Important design and applicability criteria for benchmarks include relevance and representativeness for the field, fairness, repeatability, cost-effectiveness, scalability, and transparency [3]. One caveat is that vendors may optimize hardware specifically for these benchmarks, potentially neglecting new real-world problems and emerging challenges not captured by traditional benchmark suites.

Therefore, it is essential to provide a diverse set of benchmarks so that different communities can evaluate and interpret results in terms of the performance metrics most relevant to their specific needs.

HPC benchmarking has traditionally focused on supercomputing performance comparisons, targeting compute performance [4, 5], as well as memory, communication, and storage performance [6, 7]. With the resurgence of AI and machine learning—including deep learning—it is now appropriate to explore additional lessons for benchmarking drawn from these domains.

HPC benchmarks are often executed under controlled conditions, such as those maintained by system administrators, to ensure exclusive access to hardware and eliminate interference from other users or applications. This approach allows for measurement of the best achievable performance and is frequently used to guide system procurement decisions. However, such conditions do not reflect the shared nature of most computing environments, which often include factors such as queue wait times and concurrent multi-user workloads sharing hardware resources.

2.3 What is Democratization?

We believe it is vital not only to allow experts and power users to participate in benchmarking efforts but also to lower barriers to entry — making powerful benchmarks, tools, knowledge, and infrastructure available to everyone, not just those with specialized resources or expertise. For benchmarking, this implies in particular to improve the following:-

a. **Accessibility:** Making benchmarks easier to use, enforcing open-source licensing.
b. **Open participation:** Encouraging community contributions through open-source development (e.g, on GitHub; shared repositories with transparent governance).
c. **Knowledge sharing:** Providing tutorials, documentation, and educational resources so that non-experts can effectively use and modify the benchmarks.
d. **Affordability:** Reducing cost barriers not only by introducing open source benchmarks, but also by allowing benchmarks to be offered at various scales and not only for leadership-class computing resources.

2.3.1 AI Software Democratization

One of the major success stories in the field of artificial intelligence is the emergence of AI-specific software libraries such as TensorFlow, PyTorch, and Jupyter Notebooks. These tools have democratized machine learning and data science by making advanced computational capabilities accessible to students, researchers, and small organizations that previously lacked the resources to develop such tools from scratch.

2.3.2 AI Hardware Democratization

One must recognize that a significant amount of progress in AI research is conducted on campus computers that are much smaller than hyperscale AI machines or leadership-class government systems. Furthermore, many scientists have begun to use *desktop* computers equipped with high-powered graphics cards. Hence, it is important to have meaningful AI benchmarks available that allow for comparisons across different scales.

2.4 What is Software Carpentry?

To set the stage for why we need AI benchmark carpentry, we need to first look at how the term has been introduced and is now commonly associated with software carpentry. After a more detailed analysis of software carpentry, we define the term AI benchmark carpentry.

Software Carpentry [8] was initially conceived to teach researchers in scientific fields fundamental computational and software development skills, analogous to a hammer or level in a tool belt. Thus, non-computer scientists would be able to improve the use and development of the software they need to conduct their own research while benefiting from targeted, short educational tutorials.

Today, a global community effort has sprung up since 1998 [9] that provides a number of training materials and sessions to the scientific community to we can leverage in some extend. Recently, additional areas beyond software, such as data carpentry. Together, these efforts includes:

- **Software Carpentry Core Efforts:** Teaches researchers foundational computing skills to enhance their productivity and efficiency in research tasks. This includes lessons in Programming with Python, Version Control with Git, The Unix Shell, Programming with R, Python, and using Git for version control.
- **Data Carpentry Efforts:** Teaches researchers skills necessary to work effectively and reproducibly with data in the context of specific domains. This includes lessons in the fiels of Astronomy, Ecology, Genomics, and Social Science with crosscutting topics such as Geospatial and Image Processing. Within those areas, are lessons such as Data Analysis and Visualization in R for Social Scientists, Foundations of Astronomical Data Science, and Introduction to the Command Line for Genomics [10].
- **Other Carpentry Efforts:** Library Carpentry provides lessons for information scientists, data stewards, and roles in library science, reusing some of the Software Carpentry topics adapted in a curation context. Additional lessons available include High-Performance Computing (HPC Carpentry) [11, 12].

From this list, we see that benchmark carpentry is missing.

2.5 What is Benchmark Carpentry?

Based on our observations in the educational and scientific communities [13], we find that similar efforts are needed to focus on benchmarking. This is more important as AI applications consume enormous resources, and properly scaling and using them requires a much deeper understanding of their time and space requirements. The hope is that, from similar benchmarks, not only can the scientist learn lessons about their own applications, but, if needed, their own benchmarks can be developed to estimate costs and effort more precisely. In addition, reproducible, portable benchmarks enable the selection and comparison of suitable hardware for the effort.

In general, we distinguish between hardware, software, and application components that significantly impact benchmarks.

On the hardware side, we deal with compute-oriented components such as CPUs, GPUs, and/or AI/neural accelerators (NPUs). Benchmarking them in the traditional way includes processing speed, core utilization, and instruction efficiency of a computer's central processing unit, data movement between xPU and main memory, to name a few. However, for AI, we also need performance in parallel computation, as well as AI workloads derived from AI kernels and applications.

As many AI applications require a large amount of *data* to be moved between memory, disks, CPU, and GPU memory, evaluating bandwidth, latency, and throughput is critical to understanding their impact on system performance. Hence, estimating and measuring the impact of, for example, assessing read/write speeds, IOPS, and access latency to identify bottlenecks in data storage systems is important.

Related to this is the *Network performance* metric, which measures bandwidth, latency, and packet loss to ensure efficient data transfer across systems, especially when parallel processing is used to address the scale required for good performance.

Benchmark carpentry should also teach *System Profiling and Monitoring* principles and tools so as to measure real-time system metrics. *Interpreting Results, Analyzing Bottlenecks*, and *Optimizing Performance* are essential skills to identify limitations and improve overall performance through iterative strategies. *Benchmark Design and Reproducibility* are similarly essential to allow comparative analyses among heterogeneous and also decentral benchmark runs. This includes fair, repeatable benchmarks that reflect real-world workloads and enable comparative analysis of the different components involved.

3 TOWARDS A FORMAL SPECIFICATION FOR AI BENCHMARKS

As part of the MLCommons Science Working group meetings, we have identified that ingredients of ML benchmarks include:

1. Datasets (such as images, application specific scientific data, time series)
2. Tasks to be performed
3. Methods to perform these tasks (such as machine learning models, language models)
4. Metrics (runtime; accuracy; efficiency computed from the resources required for executing the task, such as space, memory usage, energy efficiency, power draw)
5. ML oriented performance impacts such as Latency impacted by the time per inference, Throughput for the inferences per second, and training time to reach target accuracy.
6. Replication which includes the ability to replicate the experiment while at the same time being able in a structured fashion to compare the results.

3.1 Formalization

To formalize the specification of a benchmark we introduce the following notation

$$B = (I, D, T \textit{ or } W, M, C, R, V)$$

$$\begin{aligned}
B &= \text{Benchmark} \\
I &= \text{Infrastructure} \\
D &= \text{Dataset} \\
T, W &= \text{Scientific Task or Workflow} \\
M &= \text{Metrics} \\
C &= \text{Constraint} \\
R &= \text{Results} \\
V &= \text{Version or Timestamp}
\end{aligned}$$

Further we define the task to be executed as an application applied to a set of parameters.

$$T = (A, P)$$

$$\begin{aligned} A &= \text{Application} \\ P &= \text{Parameters} \end{aligned}$$

Alternative to a task, a workflow W can be used, if it contains multiple tasks that need to be conducted to achieve the scientific task (see Section 3.4).

Each of B, I, D, T, M, R, A can have constraints C_c, where

$$c \in \{B, I, D, T, M, R, A\}$$

In case of static benchmarks, many of the parameters may be fixed. However, when defining dynamic benchmarks, we define a metric that is to be minimized while allowing a predefined set of parameters of the benchmark to be variable. Let $B_i(M)$ denote a benchmark with a fixed metric M and variations in I, D, T, C, R specified by i. We try to identify the minimum

$$min\{B_i(..., M, ...)(S_j) \mid \forall_j M(S_j)\}$$

where $M(S_j)$ is the value of the solution for the metric and S_j identifies a solution parameter set for the given metric. Please note that due to the statistical nature of the AI algorithms used in the benchmark, multiple solutions exist. However, we are not suggesting to conduct an exhaustive search of all possible solutions.

Let us assume M denotes the scientific accuracy of the benchmark; then, we look for the best scientific solution. Frequently, other restrictions are applied to the benchmark to make it tractable. While it is common to restrict the dataset, variation of the tested algorithm (the function we minimize) is often desired, since the scientific community is often not only interested in comparing hardware, but also in finding the best algorithmic solution. Such a solution can then be further studied with respect to efficiency or cost metrics.

Next, we briefly describe each of the parts that comprise a benchmark in more detail.

3.2 Infrastructure

Infrastructure refers to the computational and software environment required to execute the scientific task.

This includes computational hardware, software libraries, operating systems, and cloud platforms, but also power related infrastructure to operate the resources. In many cases some of these parameters are targeted by the benchmark for comparison (e.g., different types of GPUs). As a guiding principle, an attempt should be made for each single benchmark to be clearly described with as many infrastructure parameters as possible. This will foster a clear description, reproducibility, and comparability of the benchmark.

Clearly defined infrastructure will help with (a) reproducibility, as it ensures results can be reproduced across different environments, (b) fairness, as it identifies clearly the differences between different hardware and software used, (c) scalability as through comparison we can identify various scalability issues and properties, (d) efficiency, as we can assess resource use in regards to common metrics such as time, space, energy, and cost.

3.3 Dataset

A dataset or multiple datasets provide the input data for the scientific task to be performed. Datasets in benchmarking need to be stratified into training data (used to develop a machine learning model by direct interaction with the data), validation data (used to develop a machine learning model by indirect interaction, i.e., hyperparameter tuning), and test data (used to evaluate machine learning model performance after training). If the benchmark is concerned with hardware performance, not training any machine learning model, only a test dataset might be needed. In many cases, it is important to provide different sizes of data sets to enable (a) a small set for fast development of the approach, and (b) a larger set that fosters scientific accuracy with longer run-times. Intermediary sizes are also sometimes needed to adapt to available resource constraints to compare them on different scales. Data should always be sufficiently described through metadata or documentation so their context within the scientific application can be determined. Together, these facilitate the establishment of (a) a ground truth that serves as the basis for evaluating scientific accuracy (b) a relevant and representative example that is influential for the scientific application, and (c) the identification of bias for data-driven applications.

We distinguish two different data sets: static and dynamic. If behavior can be tested statically, this is to be preferred; introduction of hyperparameters into a testing setup results in combinatorial explosion of possibilities, making some benchmarking approaches intractable or prohibitively expensive. In such case, constraints could be posed to restrict the benchmark to the most meaningful hyperparameters. In fact, doing this as part of the workflow could be an integral part of the benchmark. For instance, a standard runtime test of a given compute task on different GPUs does not require dynamic datasets, as it is not expected that the results will change over time; the hardware parameters are fully specified. Recent efforts have shown that, in some cases, we need to consider live data ingestion into benchmarks, for example, in earth science or health care applications, to support real-time predictions. We term such datasets *living datasets*, which are continuously updated with new data, edge cases, or corrections. Such living data sets are a special case of dynamic datasets. Such living datasets could be real-time data, but they could also be simulated using a static dataset while ingesting the data over time. While modifying the dataset the benchmark could evolve over time as the data available may be growing or become more accurate, supporting the need to identify the most accurate solution.

Living datasets allow us to maintain the relevance of a benchmarking task over time while simultaneously reacting to changes in the benchmarked systems.

It can also be used to adapt the benchmark to issues like over- and underfitting.

One additional aspect is that it can be useful to simulate such datasets and observe the changes of the benchmark when such data sets are utilized. Activities such as developing digital twins promote such approaches.

3.4 Scientific Task

The scientific task identifies the core challenge being evaluated while precisely identifying the purpose of the evaluated components. Typical tasks include classification, translation, reasoning, time series prediction, and planning. Through its precise definition, it sets the scope of the benchmark and introduces the community to the task to be executed and/or measured.

In more complex situations, the task itself may be a scientific workflow comprised of interacting components. In that case we may use a graph specification of the scientific task that uses subtasks that interact through edges indicating data flows and temporal executions. In that case we can use W instead of T as the specification of a workflow with properly augmented edges. Each task could have its own benchmark.

Formally, $W = (T, E)$, where T represents the collection of all tasks

$$T = \{t_1, t_2, t_3, \ldots, t_n\}$$

where n is the total number of tasks, and E indicates the dependencies between the tasks.

$$E = \{(t_i, t_j) \mid t_i, t_j \in T, t_i \neq t_j\}$$

where $(t_i, t_j) = (t_j, t_i)$.

The introduction of Workflows into the formal definition is also motivated by the recent introduction of *Agentic AI frameworks* to support automation and benchmarking of it.

3.5 Metrics

Metrics are quantitative measures used to assess the relative performance of the tested system in completing the scientific task. It has been shown in much previous work that the selection of the metric is the most crucial part of the benchmarking process.

The choice of metric determines many other aspects of the benchmarking purpose. For instance, by choosing runtime (e.g., wall clock time) as the main metric, it is strongly implied that the benchmark's main purpose is to find the fastest hardware or algorithmic implementation. By choosing an accuracy metric (e.g., F1 score), it is instead implied that the predictive performance (e.g., in classification tasks) is the target of the benchmark. Complex metrics can visualize trade-offs between the primitive metrics; for instance, a benchmark for the efficiency of a classification algorithm can weight its F1 score against the runtime (per sample inference speed), model size (in parameters), and energy requirements.

Implemented in this way, metrics can be used to establish a ranking of the benchmarked components, given they were measured in similar circumstances and under similar constraints.

3.6 Constraints

In many cases, it is necessary to constrain the benchmark to make the comparison tractable. This may include limits to training, inference, model size, or the amount of data used. Introducing constraints can (a) improve fairness while executing the benchmark (b) address operational real-world limitations, and (c) simplify the experimental setup. Constraints can be applied to any component of the benchmark, e.g., C_I, C_D, etc.

3.7 Results

A benchmark must produce clear easy to comprehend results to allow evaluation of the task performed and to perform unambiguous performance evaluation. As described above, a major determinant of the informativeness of a benchmark is the choice of metrics. Performance can be evaluated on main metrics (e.g., accuracy or runtime), but often also includes a grid search of various methods, models, and hyperparameters. To simplify comparison, metric dashboards with charts and tables, as well as error analysis, are recommended. This allows (a) analysis of progress over time, (b) informing stakeholders about model capabilities, (c) identifying limitations of the tested methods, and (d) establishing a potential leader board for selecting suitable candidates that may be applicable to similar scientific tasks.

4 REVIEW OF BENCHMARK RELATED TO THIS EFFORT

This section provides an overview of key benchmarking efforts that motivated our paper. We start with HPC benchmarks and also address MLCommons benchmark efforts.

4.1 HPC Benchmarking

HPC benchmarking has a great impact on the activities that we report here and we can learn a lot from these efforts. Some of the most known efforts are TOP500 and Green500.

4.1.1 TOP500

The list of world's largest supercomputers has been released biannually for nearly 4 decades now and thus offers a number of important lessons in designing sustainable benchmarks. At the heart of the TOP500 scoring procedure, which yields a ranked list of 500 supercomputing installations, is the LINPACK benchmark [14], which bears the name of the namesake software library [15] for solving systems of linear equations. This linear solver package was designed in the 1970s and implemented in FORTRAN. The user guide for the library was published in 1979 and included a list of only 24 computers [15]. The following decades brought in various aspects of scaling into the software, the list sizes, and the machines submitted for inclusion in the ranking as well as data and reporting information.

4.1.2 Green500

Power and energy play a dominant role in the modern world of high-performance and distributed computing, with multi-megawatt data centers and computing facilities abound in many locations across the globe. The issues of excessive power draw and energy consumption data in the mid-2000s [16, 17] culminated in a special working group of cross-industry members [18, 19], combining the TOP500 ranking with the available power draw information from the supercomputers to yield the ranking called Green500 [20]. Since then, it is published alongside the TOP500 ranking and continues to underscore the importance of efficient energy use at large HPC installations.

4.1.3 HPC innovation

Besides the recognition of development of tools and software to facilitate the use of HPC systems and foster democratization, power consumption monitoring has been integrated at the various levels of HPC facilities, from the processing and networking elements to the data center level infrastructure. Also, by utilizing different floating-point precisions [21] the applications improve their efficiency and benefit from a great impact on the system performance due to direct targeting of the specific architectural designs.

The creation of leaderboards has led to a better understanding of the overall HPC system, but insights can be limited by misalignment of algorithm scaling and leaderboard projections. To counter misalignment, benchmarks should closely resemble the scientific task to be benchmarked. In some cases, it is informative to include end-to-end performance, including data storage limitations.

4.2 Machine Learning Benchmarks

Benchmarking in scientific machine learning (ML) has emerged as a critical area to guide algorithm development, enable fair comparisons towards progress and innovation, and facilitate reproducibility. The development of ML benchmarks for science is especially critical because of the multi-disciplinary nature of the development, often including domain experts, computing hardware developers, and ML researchers. That, coupled with the variety of tasks and workloads, makes *high quality* benchmarking critical to making progress.

To obtain an overview how many academic benchmarks have been published in well known public domain archives, so we queried arXiv [22] and Google Scholar [23]. Note that according to Google, Google Scholar does not include all entries from arXiv, but it does include most of them. However, it also includes many more resources, so we expect a larger number from Google Scholer. As of Oct 1, 2025, we find 106 entries on arXiv when searching for the topic *"AI benchmark"*. executing equivalent queries in Google Scholar yields 2,490 entries for *"AI benchmark"*. It is evident from this that a complete survey of these papers is difficult to achieve through manual inspection. In an upcoming effort, we plan to explore how to automatically categorize these entries using LLMs while implementing an agentic AI framework for it.

The vast number and diversity of scientific tasks poses challenges to finding a well-defined, high-quality benchmark for any given task. To improve discoverability, we have cataloged in this paper all MLCommons benchmarks that have a result submission. Secondly, we have developed an ontology [24, 25] that allows users to identify suitable benchmarks.

4.2.1 MLCommons

MLCommons [26] provides one of the most comprehensive and standardized ecosystems of AI benchmarking. It addresses training, inference, scientific computing, and domain-specific benchmarks. Most prominently, the MLPerf benchmark suite—covering datacenter, edge, mobile, and training applications—establishes industry-wide baselines for performance, accuracy, power efficiency, and quality of service across diverse model classes such as computer vision, language, recommendation, speech, and reinforcement learning. Additionally, it offers specialized evaluations including MLPerf Tiny for microcontroller-class devices, MLPerf Storage for I/O workloads, and MLPerf Science for large-scale scientific AI. Furthermore, MLCommons promotes the reproducibility through initiatives such as Croissant ML, a standardized metadata schema for datasets, and MLCube, a portable

Table 1. MLCommons Benchmarks

This table and the refernces included in that table are located in the supplementary document.

Table 2. Ontology Table for Selected AI Science Benchmarks.
(For detailed view of the Radar Charts, see [24].)

This table and the refernces included in that table are located in the supplementary document.

container-based model packaging standard. Additional domain-specific working groups in medical AI, multilingual speech, and responsible AI have recently expanded the targeted domains.

We have provided a comprehensive list of benchmarks in Tables 1 and **??**. The tables contain information about the benchmark name, model, task, domain, model type, metrics, hardware, and a brief note. The evaluations of the AILuminate benchmarks can be found on the MLCommons Web pages and include (a) Safety / Jailbreak Tests, (b) LLM Safety Evaluation, (c) Responsible AI / Alignment (d) LLM (Decoder) (e) Safety Rate, Toxicity Score (f) Cloud LLM APIs (g) Robustness and Alignment.

4.2.2 Ontology

To improve discoverability of suitable benchmarks for a given task, we introduce a definition and AI Benchmark ontology of scientific machine learning benchmarks, where benchmarks are classified and mapped to their scientific domain and machine learning task type in [25]. This work grew out of the Web page created at [24], [27] and provides an easy to use interactive mechanism to query the cataloged benchmarks.

New AI benchmarks are added through an open submission workflow overseen by the MLCommons Science Working Group. Each submission is evaluated against a rubric of currently six categories (Software Environment, Problem Specification, Dataset, Performance Metrics, Reference Solution, Documentation) that assigns an overall rating and potential endorsement. The scoring framework enables stakeholders, researchers, domain scientists, and hardware vendors to identify representative subsets of benchmarks that align with their specific priorities. The ontology supports adding new scientific domains, AI/ML motifs, and computing motifs.

A subset of information collected by the Web page is shown in Table 2. It not only includes some elementary information about the benchmarks but also a perceived rating displayed as a radar chart. Such radar charts include ratings from 1-5, where 5 is the best rating. Ratings are identified for documentation, specification, software, metrics, dataset, and reference solution. The Web page not only includes an automatically generated report of all benchmarks in PDF format, but also a convenient online publication of the benchmarks with convenient search capabilities.

4.3 Technical aspects of AI Benchmarks

In addition to discoverability challenges, there are also technical issues that need to be addressed in dealing with democratization and AI benchmark carpentry.

4.3.1 Workflows

There are many workflow frameworks that can support the AI Benchmark Workflow. Two of them are the Compute Coordinator and the Experiment Executer; they can be used in conjunction or separately [28]. The Compute Coordinator allows hybrid infrastructure access from the benchmark application, while the Experiment Executor allows the repeated execution of templated benchmarks. Both produce results in a structured fashion so they can be combined from multiple experiments and multiple infrastructures in order to support the FAIR principles.

4.3.2 Containerization

Benchmarking on HPC and even smaller machines can be simplified by providing containerized environments which not only enable easy deployment, but also can harmonize execution by providing stable operating system and software environments. In addition to portable makefiles, the uniform generation of containers can be leveraged between applications. Although docker is today widely used to containerize applications, on HPC systems we find that limited root access on many HPC systems led to the development of apptainers. Hence, AI benchmarking carpentry should include the development of software in apptainers directly or converting Docker containers to apptainers.

4.3.3 System-Dependent Software and Deployment Variability

Benchmarking can be complex if the software, libraries and infrastructure differ across systems. To support coordinated benchmarking across different machines, we have introduced a templates hybrid reusable computational analytics workflow management framework with cloudmesh. This framework has been be applied to multiple Deep Learning MLCommons Applications. The details are explained in [28]. Utilizing such workflow systems promotes adaptation as deployment and execution is typically included in the workflow specifications. However, it can also address adaptation and modifications to future improvements and porting to different hardware as a working template is already provided..

4.3.4 Logging and Monitoring

A variety of logging frameworks exist for AI Benchmark logging. This includes logging tools such as MLPerf logging. While such tools provide elementary logging features, their outputs are not human readable and require post processing. This is also an issue when running applications in interactive mode during debugging phases. For this reason, we have provided Cloudmesh-stopwatch that not only allows human readable format, but also allows automatic MLPerf logging (if desired) with a single line change in the code. Cloudmesh stopwatch supports Python, shell, and batch script execution, and employs a consistent log format across all three.

In general, we distinguish between four types of monitoring: (a) Infrastructure Monitoring, (b) Application Monitoring, (c) Training Monitoring, and (d) Model-Level Monitoring. A wide range of tools exists for each type, making it essential to identify those that provide effective functionality while remaining easy to use. TensorBoard is one example.

4.3.5 Profiling and Performance Analysis

Profiling is the process of measuring a program's performance in association with the locations in the source code in order to reveal where resources (e.g., time and memory) are spent during execution. Profiling is important in AI benchmarking for the following reasons:

Table 3. Summary of Example Profiling Tools Useful for Deep Learning and AI Workloads

This table and the refernces included in that table are located in the supplementary document.

- Profiling helps explain why a particular method or implementation variant is faster than another.
- Profiling helps support fair and reproducible benchmarking.
- Profiling can distinguish between the essential computations and extraneous overheads.
- In a heterogeneous system, profiling can identify which components (e.g., CPU or GPU's CUDA cores vs. tensor cores) are being used by different parts of the application.
- Profiling can identify which specific library kernels are being used by different parts of the application.

Table 3 provides a list of profiling tools that are useful for analysis of deep learning applications.

It is important to note that the tooling and services exist for supporting different levels of infrastructures. This includes examples for framework-level, system-level (including CPU and GPU), kernel-level, compiler-level, communication-level, and cloud-level.

Furthermore, we aim here to provide comprehensive coverage of the AI profiling stack, which affords users the insights into cross-vendor and cross-platform capabilities and offerings, and also provide key analysis of features of the said tools and services.

We believe it is essential to increase awareness and use of profiling tools through AI benchmarking efforts, enabling a better understanding of bottlenecks in AI applications. Additionally, we need to educate the community about policy limitations that may implicitly restrict specific profiling tools. As discussed previously, one such policy restriction is that not all profiling information is available for energy benchmarks. Such restrictions may also be in place for additional hardware profiling measures.

Lastly, we need to educate the community about the *performance impact* of profiling costs to avoid over-profiling. Therefore, it makes sense that AI benchmarks should be able to choose the level of profiling selectively. This information is vital to support the FAIR principles and ensure that benchmarks are comparable.

4.4 GPU Benchmarking and its Variability

Modern scientific applications frequently require peta- or exascale levels of compute to model topics with high fidelity. To meet these demands in reasonable timeframes, scientists and researchers typically run these workloads on massively parallel systems such as GPUs. For example, workloads such as graph analytics [29, 30], scientific computing [31, 32, 33, 34], ML [35, 36, 37, 38, 39, 40, 41, 42] heavily utilize GPUs. Increasingly, ML is also impacting scientific applications [43, 44, 45, 46, 47] by replacing or supplementing traditional computing methods in application domains like molecular dynamics (e.g., DeePMD [48, 49]), protein folding (e.g., OpenFold2 [50]), and scientific AI models (e.g., AuroraGPT [2]). However, given the scale of data these workloads operate on and the large size of the workloads themselves, they typically must partition their work across many GPUs.

Given their widespread use and trend towards many GPU applications, it is desirable from a benchmark carpentry perspective to make GPU experiments repeatable and consistent. For traditional HPC systems composed of multiple CPUs, prior work showed that this was difficult to achieve: application performance varied by up to 20%, even for CPUs with the same architecture and vendor SKU (Stock-Keeping Unit) [51, 52, 53, 54, 55, 56]. This variation occurs due to the manufacturing process and the chip's power constraints [53, 57]. Such dynamic behavior makes it challenging for repeatable experiments, and can lead to resource underutilization. Unfortunately, similar issues also arise in modern systems composed of many GPUs. Recent work has demonstrated that GPU-rich systems suffer from significant performance variability [58, 59, 60, 57, 61, 62, 63, 64, 65].

For example, Sinha, et al. examined variability across five modern GPU-rich clusters with a variety of sizes, cooling approaches, and GPU vendors [62]. They found that applications exhibited performance variability of 8% on average (max 22%) with outliers up to 1.5× slower than the median GPU. Moreover, these results were consistent over time (i.e., not transient) and were unaffected by GPU vendors or cooling type. Interestingly, this performance variability was also application-specific: the more compute-intensive the application was, the more performance variability the application observed due to effects of the GPU's power management algorithm (e.g., Dynamic Voltage & Frequency Scaling—DVFS). Furthermore, performance variability is getting worse as transistors continue scaling [66].

Although the impact of performance variability is significant for single-GPU workloads, it is even larger for multi-GPU workloads. Currently, GPU-rich systems focus on scheduling work to minimize the number of nodes an application requests, without considering variability. In the five clusters from this prior work, users asking for 4 GPUs for a given application would get a slower GPU allocated to them between 22% (Sandia's Vortex cluster) and 50% (TACC's Longhorn cluster [67, 68]) of the time. Thus, users are likely to get a slow GPU frequently, especially since modern scientific workloads often request 64 or more GPUs for a given experiment. This can lead to significant resource under-utilization for multi-GPU jobs since all of them must wait for the slowest one to complete due to the bulk synchronous programming (BSP) model used in many data-parallel workloads [69]. Accordingly, it is imperative for users to be aware of the impact of performance variability on their experiments, and for benchmark carpentry to propose solutions to minimize its effects.

Although GPU-rich systems are likely to suffer from performance variability for the foreseeable future, there are several steps various stakeholders, such as users, maintainers, and system designers, can take to reduce the impact on obtaining statistically significant results in existing systems. First, cluster operators can perform periodic performance-variability benchmarking to identify underperforming GPUs and perform targeted maintenance on them. Likewise, users can perform similar benchmarking to identify GPUs that behave similarly, and then use blacklisting or other scheduling approaches to attempt to schedule work on GPUs with similar performance variability profiles. However, doing so can be time- and labor-intensive for clusters with thousands or more GPUs (though it is a one-time cost, since a GPU's performance variability is consistent over time). Thus, a more scalable, dynamic approach is to redesign job-scheduling policies for GPU clusters to account for performance variability when making scheduling decisions. Recent work has shown that embracing performance variability can transparently and significantly improve job completion time, makespan, and GPU utilization [70]. Finally, since performance variability is application-specific, we recommend that new, unprofiled applications either rely on other applications with similar

Table 4. Estimated Energy Consumption of GPT Models for Training and Inference

Model	Training Energy (MWh)	Inference Energy (per 1M queries, MWh)
GPT-3	∼1,287 [79, 77]	∼50–100
GPT-4	51,773–62,319 [80, 81]	∼600–1,000
GPT-5	>60,000 (estimated) [82, 83]	∼800–1,200
GPT-6	80,000–100,000 (projected) [82]	∼1,000–1,500

profiles as proxies [71] or be profiled during their first execution on a new cluster to determine their sensitivity to performance variability.

In terms of democratizing the availability of multi-GPU systems there are several barriers to overcome, these are the cost, access, skills and complexity The cost barrier means that the large-scale systems are affordable only to national labs and major corporations. Consequently, the access is usually restricted to the staff of these organizations. Using multi-GPU systems effectively requires specialized knowledge. Users must be trained in containerization technologies, distributed libraries, and orchestration tools that allow applications to scale across many GPUs. There is also a barrier on the conceptual level. The performance of a multi-GPU system is the result of interactions between hardware, interconnects, and software stacks. At present, we lack high-level performance prediction model that can reliably describe how applications behave when running on GPUs. This makes it difficult to plan experiments, determine the required resources and generalize findings.

4.5 Energy Benchmarking

Energy consumption is a critical component of ML benchmarking. Training and inference with modern AI systems can require enormous computational resources.

To illustrate the issue, we have provided in Table 4 and Figure 1 the energy required to train various ChatGPT models (some of which are estimated as no public data has been released [72, 73], such as GPT-5 and GPT-6). The training of a single large-scale language model (GPT-3) consumes approximately 1,287 MWh placing it in the same range as the annual energy usage of about 130 U.S. households, according to U.S. Energy Information Administration (EIA) statistics on average residential electricity consumption [74, 75, 76, 77, 78, 79].

For the U.S. Department of Energy (DOE) leadership-class machines, such as those hosted at Oak Ridge National Laboratory (see Table 5), we find documented and significant progress toward exascale, but at the cost of increased energy consumption that more than doubled during the last generational upgrade. However, the Peak Performance per energy unit has increased significantly, and compared to Jaguar's initial values, Frontier has improved by a factor of 209, thus becoming relatively more efficient despite overall energy consumption needs.

Carbon-emission measurements also help provide a more detailed understanding of associated energy impacts.

If we only focus on traditional benchmarks using metrics such as FLOPS or latency, we provide performance insights but overlook *energy-to-solution*, which measures the total energy required to complete a task. Without perspective, researchers and practitioners focus on optimizing for speed at the expense of sustainability and cost efficiency.

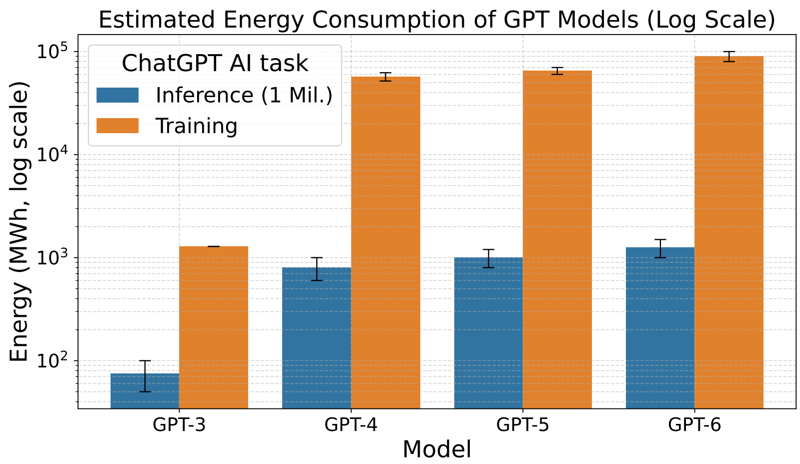

Figure 1. Energy Consumption for ChatGPT Training and Inferencing 1 Million Queries. (Data for GPT-5 and 6 are estimates).

Table 5. Evolution of the Leadership Class Supercomputer at Oak Ridge National Laboratory

Machine	**Year**	**Architecture**	R_{max} Scaling	R_{max} **PFlops/s**	R_{peak} **PFlops/s**	**Power (MW)**	R_{max}/**Power (PF/MW)**
Jaguar[84]	2009	Multi-core CPU	1	1941	2628	7	277.29
Titan[85]	2012	Hybrid CPU/GPU	9.06	17590	27113	9	1954.44
Summit[86]	2017	Hybrid CPU/GPU	76.6	148600	200795	13	11430.77
Frontier[87]	2022	Hybrid CPU/GPU	697.1	1353000	2055717	29	46655.17

*PF=*Theoretical peta–floating-point operations per second*; 1 PF = 10^{15}FLOPS.
R_{max} = maximal LINPACK performance achieved. R_{peak} = theoretical peak performance.

Thus, we believe it is important to make energy benchmarks an important aspect of AI benchmarks. Energy benchmarking ought to address the following:

- Quantify the environmental footprint of AI workloads (carbon emissions, renewable vs. non-renewable energy use).
- Highlight economic tradeoffs in large-scale computing (cloud costs, datacenter efficiency).
- Guide hardware and algorithmic choices towards a more effective architecture.
- Support policy and funding decisions by providing transparent data on sustainability.

Energy-aware benchmarks help ensure that AI development aligns with broader goals of responsible computing, making results reproducible, performant, and economically and environmentally sustainable.

Thus, we see several opportunities. First, we need to make energy benchmarks more prominent and provide materials and tutorials as part of AI benchmark carpentry to educate the community. Second, we must ensure that not only the most expensive hardware, such as leadership-class and hyper-scale data centers, is used, but also medium- and even small-scale hardware, so that democratizing energy benchmarks within the community is easy to implement. This way, measurements of even smaller AI-based scientific applications can integrate energy consumption into their benchmarks, and meaningful comparisons with traditional algorithms that do not use AI can be drawn. Third, we must ensure that energy metrics and logs can be accessed and uniformly integrated into the AI benchmarks.

4.5.1 AI Energy Benchmark Carpentry

To support AI energy benchmark carpentry efforts, we need to address the following issues:

- Conduct a relevant survey of existing efforts
- Identify metrics useful for AI benchmarks
- Identify how to leverage existing and create new leaderboards focusing on energy metrics
- Identify simple-to-use blueprints as part of the carpentry efforts that can not only be replicated and reused, but also serve as a basis for newly developed benchmarks.
- Conduct community outreach to offer carpentry tutorials that focus on AI benchmarks instead of just AI software and services.
- Identify how to obtain and integrate meaningful and practical metrics (e.g., data centers may not provide uniform access to energy data) so that energy data collection and access become part of carpentry efforts.

Strategies to integrate energy into AI benchmarks for carpentry efforts include improving access to metrics, including the creation of logs during runtime that:

- Log ambient temperature and humidity.
- Log sample power at regular intervals or averages over the run.
- Store the logging data in an easy-to-parse format (CSV, JSON, YAML)
- Upload results as artifacts in support of the FAIR principle and make available for comparison.

Next, we discuss some of the aspects that need to be addressed in more detail.

4.5.2 Energy Metrics

There are various energy metrics to consider, including metrics that may not historically received attention. It is also important to identify metrics for leaderboards, but they must be obtained in a way that allows fair, informed comparisons. Hence, it is important to document how the experiment should be conducted rather than just referring to the metric. In principle, blueprints should be used and adapted to make comparisons across hardware and software easier. Energy metrics are used across different layers of the AI benchmark infrastructure, which is similar to classical HPC infrastructure. We provide an example of using different metrics on the various layers in Figure 2. Such diagrams should be integrated into the blueprints provided to users to simplify understanding the benchmarks' energy scope.

As part of the energy augmentation, a clear purpose for the benchmark metric should be stated. Such examples should be collected as part of the experiment's metadata so they can be leveraged and serve as a motivator for other benchmarks. In our example from Figure 2, the purpose for each metric is as follows:

1. **Device/Micro-architectural Layer** (D_L)
   - *Energy per flop* or *Energy per inference*: Measures the energy consumed to perform a single computational operation (a floating-point operation or an inference).
   - *Temperature sensors*: *Related Logging (Non-KPI): Inlet and Outlet Temperature Sensors*: Logged because *thermal headroom* directly bounds the safe *Dynamic Voltage and Frequency Scaling (DVFS)* ranges.

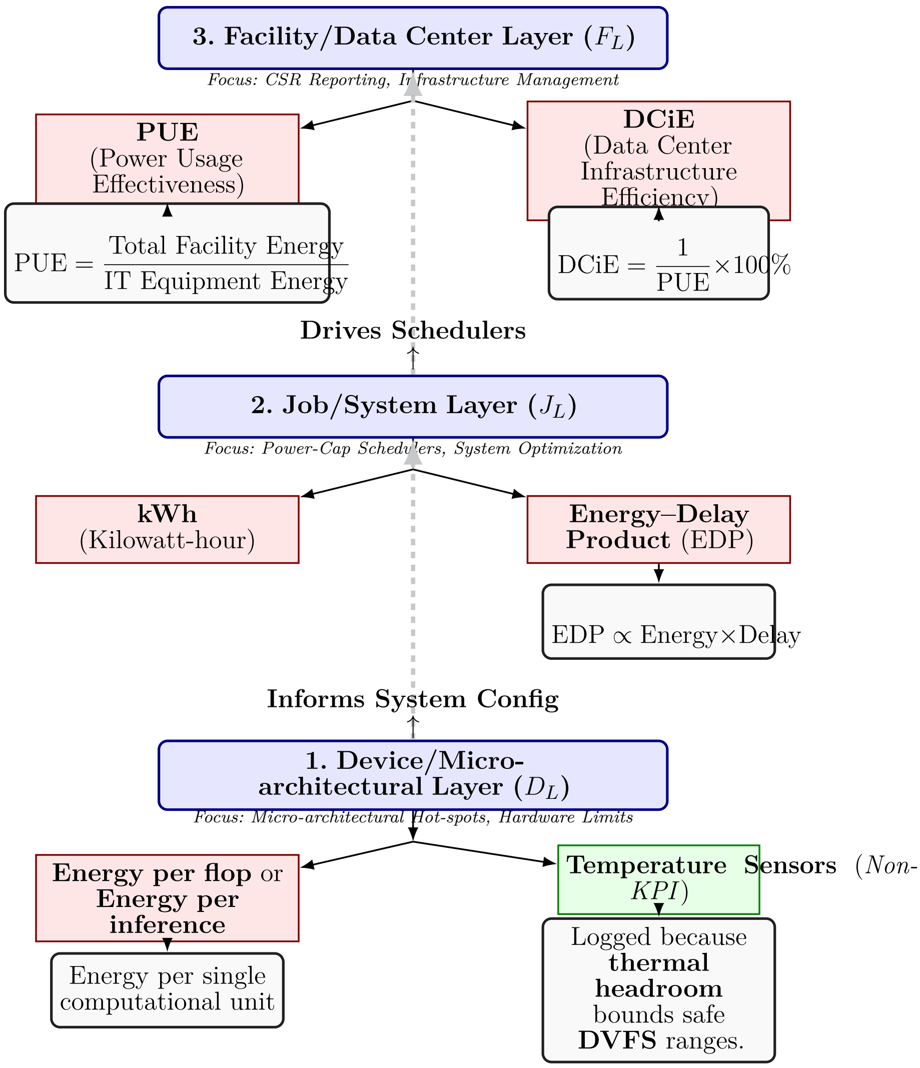

Figure 2. Illustration of an Example for Metrics as Used in the Layered System Architecture for Large-Scale AI Benchmarking.

2. **Job/System Layer (J_L)**
    - *Kilowatt-hour (kWh)*: The total energy consumed by a specific job or set of jobs over its duration.
    - *Energy–Delay Product (EDP)*: A combined metric of energy and time (energy × delay) used to assess the overall efficiency of a computation. Lower EDP generally indicates better performance and efficiency.
3. **Facilities/Data Center Layer (F_L)**
    - *Power Usage Effectiveness (PUE)*: A ratio that measures how efficiently a data center uses energy. An ideal PUE is 1.0 (meaning the IT equipment uses all energy). *Data Center Infrastructure Efficiency (DCiE)*: The reciprocal of PUE, expressed as a percentage. It shows the percentage of total data center energy used by IT equipment.

This tiered structure, along with a detailed purpose statement for each metric, allows for meaningful comparisons and decision-making at every level of the computing infrastructure.

To identify commonly used metrics, we conducted an initial survey of tools and benchmarks related to energy, which we present in Table 6, while listing their typical benchmark use.

Common requirements for such metrics include obtaining measurements at low cost, sharing results with metadata augmentations, and integrating them into potential leaderboards. We believe we have to go beyond established leaderboards such as *Green500* and the *MLPerf Power*, which already influence processor road-maps and procurement calls [88, 89], to raise awareness of the energy impact on real-world scientific applications.

Table 6. Energy- or Carbon-Efficiency (B)enchmarks and (T)ools used in Scientific-HPC research.

	(B)enchmark or (T)ool		Core metric(s)	Typical Benchmarking Use
Benchmark				
B	SPECpower_ssj2008	[19]	W/transaction; ops/W	Enterprise-server rankings; ENERGY STAR compliance
B	SPEC SERT^{2}	[90]	Server-Efficiency-Rating = kWh + perf	EU Lot 9 certification; vendor datasheets
B	TPC-Energy	[91]	Wh/DB phase	OLTP/warehouse energy cost studies
B	JouleSort	[92]	records/J	Storage-I/O contests; I/O-stack tuning
B	Green500	[20]	GFLOPS/W (HPL or HPL-AI)	Global supercomputer energy ranking
B	HPCG-Power	[93]	GFLOPS/W (HPCG)	Memory-bound tuning; procurement add-on to TOP500
B	HPL-MxP (HPL-AI)	[94]	mixed-precision GFLOPS/W	GPU/TPU evaluation for AI-optimised LINPACK
B	MLPerf Power	[95]	J; avg W; J/sample; J/epoch	Official energy track for MLPerf submissions
B	MLPerf Tiny	[96]	μJ/inference (MCU)	Edge-AI board comparison; ultra-low-power design
B	CoreMark-PRO Power	[97]	iterations/s/W (SoC)	Pre-silicon DVFS sweeps; embedded RFPs
B	UL Procyon AI Power	[98]	images/W; fps/W	Smartphone & laptop AI-inference benchmarks
B	CANDLE Power Study	[99]	J/epoch; GFLOPS/W	DOE accelerator procurement guidance
B	LULESH/miniFE Energy	[100]	J/iteration	DVFS + autotuning baselines
B	ExaSMR Power Benchmark	[101]	J/neutron; energy-vs-accuracy curve	Energy budget strategy in nuclear simulations
B	EE-HPC-WG Energy Benchmark	[102]	draft node/job spec; JSON trace	Toward common HPC energy standard
B	HPC-AI500 Energy Track	[103]	planned: GFLOPS/W; tokens/J	Mixed AI/HPC cluster evaluations
B	PARSEC-3.1 Energy Extension	[104]	W; J via PAPI-RAPL; J/op; EDP	Pre-silicon DVFS research
B	CosmoFlow-Power	[105]	J/epoch; GFLOPS/W	CNN scaling on 15 k+ GPUs
B	HACC Energy Add-on	[106]	J/particle update	N-body cosmology power studies
B	DeepCAM-Energy	[107]	J/epoch (UNet)	Climate-analytics accelerator studies
B	OpenIFS-Energy	[108]	kWh/model-day; W timeline	Weather-model node comparison
B	GROMACS-EE	[109]	J/ns; W/GPU	MD clock-vs-accuracy trade-offs
B	NAMD-Power	[110]	Energy-Delay-Product (ApoA1)	Summit node DVFS optimisation
B	QE Energy Suite	[111]	J/SCF step; GFLOPS/W	DFT GPU-offload studies
B	VASP-Power Harness	[112]	W; kWh/MD step	Materials-science accelerator compare
B	OpenFOAM-Energy	[113]	J/1k iterations	CFD partitioning & mesh tuning
B	InSAR-AI Power Kit	[114]	J/satellite scene	Edge-to-cloud EO inference cost
B	H3D-Energy	[115]	J/hydrology timestep	Hydrology model DVFS exploration
Tool				
T	PTDaemon/SERT Energy	[116]	calibrated W; kWh (node)	Lab reproducibility; Lot 9 labels
T	Scaphandre	[117]	W; kWh (process/node, Prometheus)	Slurm dashboards; power-cap feedback
T	Kepler	[118]	W/pod; J/pod (eBPF)	Energy observability in K8s clusters
T	CodeCarbon	[119]	kWh; kg CO_2e (process)	Rapid CO_2 estimation in pipelines
T	CarbonTracker	[120]	measured + predicted kWh; CO_2e	Scheduling DL jobs in low-carbon hours
T	PowerPACK/Mont-Blanc	[121]	W; J for MPI/OpenMP mini-apps	Network-topology & DVFS studies
T	Cray PAT Energy Counters	[122]	J/function; avg W	Kernel hotspot hunting on Shasta
T	IBM PowerAPI (pmlib)	[123]	kWh (job/process)	Energy-aware scheduling on Summit
T	NVIDIA DCGM Energy	[124]	W; J (GPU) 1Hz; telemetry	GPU power-cap discovery; Green500
T	Intel VTune Power	[125]	package W; J/function	Roofline-vs-energy tuning on Xeon
T	Cloudmesh GPU	[126]	Power Draw; Temperature	Temperature and energy frequency traces

4.5.3 Leveraging Previous Work

As we can see from the table, a large number of tools and benchmarks exist, and we can leverage them to work towards a FAIR-based approach on energy benchmarks. This is all the more important when developing concise carpentry and democratization efforts. The distinction in the layered architecture for energy benchmarks also helps, as it is often not possible or desirable to address all layers at once. It is evident that energy benchmarking, in itself, is a complex research topic, and that carpentry efforts must be established to bring this knowledge forward and enhance AI benchmarks into AI energy benchmarks.

4.6 Simulation as a Tool to Benefit AI Benchmark Carpentry and Democratization

Simulating AI hardware and software infrastructures offers an opportunity to democratize AI benchmarking and impact AI development. This is especially useful for those (a) without direct access to the hardware on which the AI benchmarks run, and therefore can use simulations to estimate its behavior; and (b) planning large-scale experiments, who can use simulations to assess the impact on real hardware and infrastructure.

As part of this, recent work in the modeling and simulation community has significantly expanded users' options for studying how their ML workload optimizations affect them. Although there is a wide array of tools that can be used, we focus on four of the most popular, widely used tools: Accel-Sim [127], gem5 [128, 129], SST [130, 131], and Digital Twins [132] (see also Table 7). These tools are often used in academia, industry, and national labs because they enable high-fidelity, early-stage design exploration. Moreover, they enable users who do not have access to real hardware or are prototyping optimizations for hardware that does not yet exist to simulate the behavior of popular ML workloads while balancing performance and power trade-offs.

Accel-Sim [127]: For users interested in simulating ML workloads on modern NVIDIA (Volta through Blackwell) GPUs, Accel-Sim offers a great combination of high fidelity and usability. Accel-Sim builds upon the popular GPGPU-Sim [133], and has an integrated power model [134]. This allows users to examine power and performance tradeoffs for ML workloads.

Currently, Accel-Sim supports running ML workloads in three formats: (1) direct CUDA source code, (2) CUDA programs with library calls where the library includes the PTX for the library calls (only for CUDA 8.1 and earlier [135]), (3) and direct SASS (NVIDIA's machine assembly language) execution. As NVIDIA's libraries (e.g., cuDNN, cuBLAS) grow increasingly complex, and software like PyTorch add additional complexity on top of these libraries, the third option is the most popular as it can trace through multiple layers of software (e.g., PyTorch, cuBLAS). Moreover, to make the simulator's runtime more tractable, recent work has demonstrated how to identify and simulate a representative subset of a given workload without significantly compromising accuracy [136]. Thus, Accel-Sim is widely used by users who want to improve the efficiency of a given GPU. However, since Accel-Sim focuses on the GPU, it may not be best for users who want to study interactions with other system components (e.g., the CPU or other accelerators). Accel-Sim also does not heavily focus on the GPU cache coherence or memory consistency.

gem5 [128, 129]: The gem5 simulator is another popular tool used in computer system research to evaluate novel hardware designs. It provides a robust API for researchers to modify and extend current models and to create new models in the gem5 infrastructure. The gem5 simulator implements many models for system component including CPUs (out-of-order designs, in-order designs, and others), AMD and ARM GPUs [137], accelerators [138, 139, 140], various memories, on-chip interconnects, coherent caches, I/O devices, and many others. These gem5 models have enough fidelity to boot Linux, run unmodified workloads, and investigate cross-layer designs.

Thus, gem5 enables rapid prototyping of hardware-software co-designs across the computing stack. For example, users can prototype optimizations to the compiler, OS, or runtime in tandem with architectural changes and study the implications of their design choices. Like Accel-Sim, gem5 has an integrated power model [141] and also supports running popular ML workloads both natively and through frameworks like PyTorch – including adding support for advanced techniques to tradeoff simulation time for reduced fidelity in less important application regions [142, 143, 144]. However,

Table 7. Example Simulation Tools that Benefit AI Benchmark Simulations.

Tool/Software	Scale	Benefits	Application
Accel-Sim [127]	Single- and multi-**GPU**	High fidelity, usability, integrated power model, supports NVIDIA GPUs.	Examining power/performance tradeoffs; improving GPU efficiency.
gem5 [128, 129]	Single- and multi-**CPU, GPU, and system-on-a-chip**	High fidelity, hardware-software co-design, models cache coherence, interconnects, and memory consistency, supports accelerators and AMD/ARM GPUs.	Studying ML workload behavior across components; prototyping optimizations across layers.
SST [130, 131]	**Rack-scale** systems	Faster, scalable, models networking, utilizes analytical models.	Studying ML workload behavior in large-scale systems.
ExaDigiT [132]	**Datacenter- or supercomputer-level**	Models interactions among workloads, scheduling, power, networking, and cooling, including physical footprint.	Examining ML workload behavior at the largest scales.

gem5's support for ML workloads differs in three key ways from Accel-Sim's. First, unlike Accel-Sim, gem5's support for ML workloads spans across different types of compute devices, including CPUs and accelerators. Second, gem5 currently focuses its support on AMD GPUs. Since AMD's GPU runtime and drivers are open-source, this enables gem5 to model co-design between additional layers of the computing stack because it simulates all of those layers (unlike Accel-Sim). Third and finally, gem5 also has highly accurate models for cache coherence, memory consistency, and interfaces between components in the system like the GPU's Command Processor. Thus, gem5 may be a good choice for users wanting to study how ML workloads behave across system components or who want to prototype optimizations across layers of the computing stack. However, since many users focus on NVIDIA GPUs and gem5 currently does not support them, users deeply tied to NVIDIA's ecosystem will not find it useful.

SST [130, 131]: Accel-Sim and gem5 focus on modeling a single GPU (Accel-Sim, gem5) or a single system-on-a-chip (gem5). However, modern, large-scale computing systems frequently have hundreds or thousands of processors (e.g., GPUs) integrated together. Thus, the Structural Simulation Toolkit (SST) is a good option for users who want to study ML workloads in rack-scale systems. Instead of using high fidelity, but often slow models for components like processors (like Accel-Sim and gem5 do), SST utilizes analytical models for these components and focuses on modeling the network across many components, making it faster and scalable. However, for users who want to focus on both smaller- and larger-scale systems, both Accel-Sim [145, 146] and gem5 [147, 148] have integrated their models with SST – potentially providing the best of both worlds.

ExaDigiT [132]: To study AI workloads at datacenter or supercomputer scale, ExaDigiT provides a holistic digital twin framework that models the coupled behavior of workloads, compute, power, and cooling subsystems. Unlike simulators such as Accel-Sim, gem5, or SST, which operate at device- or node-level timescales, ExaDigiT enables large-scale modeling of system dynamics over operational timescales—capturing interactions that are difficult to observe or measure directly in production environments. This framework further provides a means to evaluate operational strategies, perform"what-if" analyses, and uncover complex, cross-disciplinary transient behaviors that emerge from the tight coupling of workloads, compute, power, and cooling.

ExaDigiT consists of three coupled modules: (1) a *resource allocator and power simulator (RAPS)* for replaying telemetry, simulating the scheduling of real or synthetic workloads, and dynamically estimating energy consumption; (2) a *thermo-fluid cooling module* for predicting pressures, temperatures, flow rates, system-level control responses, and overall power-usage effectiveness (PUE); and (3) a *visual analytics module* that integrates both a web-based dashboard and extended reality (XR) interfaces for immersive exploration of system behavior in augmented, virtual, or mixed reality.

Operating at coarser timescales than cycle-accurate simulators, ExaDigiT enables comprehensive studies of power, cooling, and scheduling interactions across the full supercomputer. It has been applied to analyze how scheduling policies influence power and cooling dynamics [149], used as a reinforcement learning environment for training optimal scheduling agents [150], and to perform "virtual" benchmarking of large-scale LLM training workloads [151].

Summary: Collectively, these simulation frameworks span a continuum of modeling fidelity and scale—from device-level, cycle-accurate simulators such as Accel-Sim and gem5, to system- and datacenter-level models such as SST and ExaDigiT. By enabling controlled, repeatable, and cost-effective experimentation, they serve to democratize AI benchmarking in the design-space exploration of emerging architectures. As AI workloads continue to push the limits of power, cooling, and scheduling efficiency, such simulation-based tools will become indispensable for evaluating new ideas before committing to physical deployment.

5 SHARING BENCHMARKS

Beyond the creation of new benchmarks, *sharing* benchmakrs is an essential aspect of benchmark carpentry. To this end, integrating the FAIR principles is of paramount importance.

Benchmark sharing is best supported through hosting the code in a public repository that provides well-documented, executable workflows, thereby enabling others to reproduce the benchmark and compare results. Standard development practices, such as using Python Notebooks or scripts in other programming languages, as well as standard libraries, are recommended. More complex benchmarks may benefit from formal build processes (e.g., using makefiles) and dependency management through package managers. Containerization offers additional advantages, simplifying configuration and improving portability across environments.

To further support FAIRness, benchmark results should include standardized metadata, facilitating consistent comparison and analysis across studies.

While existing platforms such as Hugging Face and Kaggle provide mechanisms for sharing benchmarks, results, and leaderboards, fostering community capacity to host them independently remains valuable. Initiatives such as MLCommons illustrate how communities can maintain open, transparent benchmarking ecosystems. Educational efforts could be developed to train researchers and practitioners in these practices.

Finally, with the growing prominence of agentic AI, it is worth exploring its potential for automating the benchmarking lifecycle—including benchmark execution, result generation, and report synthesis. For example, the MLCommons Science Working Group is investigating how agentic AI can be applied to scientific benchmarks, particularly those involving time series analysis.

6 TOWARDS AN AI BENCHMARK CARPENTRY CURRICULUM

Based on the lessons learned and our observations from domain experts, we have devised the following exemplary curriculum addressing AI benchmark carpentry.

- **Software Carpentry Foundational Tools and Practices:**
  Before addressing benchmark carpentry, we recommend that participants will review and learn about basic fundamental tools and practices. As they already exist as part of Software Carpentry, they can be reused. However, it may be of advantage to adapt certain aspects to explicitly utilize examples that focus on AI benchmarks and not just any arbitrary software carpentry project.
  - Programming Skills: Proficiency in Python, Jupyter Notebooks, focusing on reproducible coding practices, including documentation, and reproducibility.
  - Version Control: Git for tracking changes and collaboration.
  - Command-Line Proficiency: Unix shell for efficient data manipulation.
  - Data Management: Techniques for data cleaning, transformation, and visualization.
  - Learning from Online AI/LLM Resources: Leveraging large language models and online tutorials for benchmarking insights and guidance.
- **AI Benchmarking Fundamentals:**
  Having a basic understanding of AI Benchmarking is important for designing, evaluating, and improving AI systems. Benchmarks provide a standardized way to measure performance, compare models, and identify areas for optimization. By introducing benchmarking methodologies, examples, and metrics, participants gain the tools to critically assess AI models. Effective simple visualization practices help communicating results in a transparent, reproducible fashion related to real-world examples.
  - Benchmarking Methodologies: Introduction to frameworks such as MLPerf and AIBench.
  - Scenario-Based Benchmarks: Creating benchmarks that simulate real-world AI applications.
  - Performance Metrics: Throughput, latency, accuracy, and resource utilization.
  - Displaying Information with Graphs: Visualizing benchmark results for better analysis and interpretation.
- **Reproducibility and Experiment Management:**
  Especially for benchmarks, it is not only important to document the code, but to document the results so we enable reproducibility. This includes documenting workflows and data provenance in case prior work and data are integrated. Thus, applying the FAIR principles—making data and experiments Findable, Accessible, Interoperable, and Reusable—enhances transparency and promotes collaboration across teams and institutions.
  - Experiment Documentation: Importance of detailed documentation for reproducibility and adherence to FAIR principles.
  - Automated Workflows: Using Docker and CI/CD pipelines to automate benchmarking processes.
  - Data Provenance: Tracking data sources and transformations for transparency, traceability, and reuse.
  - FAIR: Apply the FAIR principle to AI benchmarks.

- **Ethical Considerations and Bias Mitigation:**
  It is important to address the ethical implications of conducting Benchmarks. Here, we not just focus on societal impacts, but also on the reporting of bias, fairness conducted potentially through hardware, software, and even vendor impacts.
  - Bias Detection: Methods to identify and mitigate biases in AI models and datasets.
  - Fairness Metrics: Metrics to assess and ensure fairness in AI systems.
  - Ethical Implications: Discussion on societal impacts and ethical decision-making.
- **Carpentry Principles in Practice:**
  A practical experience will be introduced to showcase the principles of AI benchmarking techniques. For this, a small, manageable datasets, and AI algorithm are used. The project may be conducted individually or in groups, while a walkthrough will also be available. An expansion to this AI-based benchmark will be the hosting and deployment of a leaderboard. Contributors can post their results in this shared leaderboard for the compute systems they have access to.
  - Hands-On Workshops: Practical sessions applying benchmarking techniques to real datasets.
  - Collaborative Projects: Group projects to foster teamwork and problem-solving skills.
  - Open-Source Contributions: Participation in community AI benchmarking initiatives.
- **Special Topics:**
  As we have seen from the previous section, several aspects have a great impact on AI benchmarking, which is so far not covered by other carpentry efforts. This includes energy benchmarking, simulation of hardware to estimate performance, and performance tuning with a focus on AI. Instead of just setting up a leaderboard through, for example, a Docker container, selected parties may have an interest in finding out more about setting up such leaderboards and hosting them.
  - Energy Efficiency: Measuring power consumption and optimizing AI workloads for lower energy usage.
  - Simulation: Using synthetic data and simulated environments for benchmarking when real data is limited.
  - Performance Tuning: Techniques for optimizing model execution, hardware utilization, and system throughput.
  - Leaderboard Management: Designing, maintaining, and validating AI benchmark leaderboards for reproducibility and fairness.
  - To provide users a starting point, presenting the community with a collection of benchmarks can be useful and has been spearheaded at [152].

From the extensive surveys and numerous examples it is important that to start one ought to begin with the most elementary efforts and grow them continuously. As such, we recommend adding specific lessons when we discover they need to be added by the community. Also, we must involve the community itself and allow for contributions of tutorials from a wide variety of groups.

7 TOWARDS AI BENCHMARK DEMOCRATIZING

Our goal is to make AI benchmarking transparent, reproducible, and community-driven. Democratization empowers a broader range of participants to contribute to and learn from AI performance evaluation.

Introducing democratization tools, datasets, and evaluation frameworks that are openly accessible and easy to use can allow anyone—from students to independent researchers—to measure, compare, and improve AI models.

One of the biggest hurdles we find is that some benchmarks, probably rightfully so, target hyperscale or leadership-class machines. However, in order to increase the community and raise awareness, smaller scale benchmarks need to be available.

As such, the following aspects can improve democratization:

- **Accessibility:**
  - Benchmarks, datasets, and tooling ought to be open-source or freely available.
  - Users may not need to rely on expensive hardware or proprietary software to participate.
  - Examples can be leveraged to develop new benchmarks. One can start with examples provided by MLCommons open datasets, pre-built benchmarking pipelines, and Jupyter notebooks with ready-to-run benchmarks.
- **Usability:**
  - Interfaces, documentation, and examples in existing efforts can serve as starting point to developing user-friendly, allowing non-experts to run benchmarks.
  - Providing automated scripts and tutorials reduces the barrier to entry.
- **Transparency:**
  - Specifying clear definitions of metrics, scoring methods, and evaluation procedures ensures everyone understands the results.
  - Improved transparency addresses the hide everything in a “black box” approach, where only insiders can interpret outcomes.
- **Community Participation:**
  - Anyone with minimal but sufficient knowledge should be able to contribute to benchmarks, improve tools, or submit models.
  - Democratization also means encouraging collaboration and reproducibility across institutions and geographies (e.g., engaging the broader community).
- **Impact:**
  - Through democratization, smaller teams or educational institutions can contribute and benefit from learning, competing, and comparing AI benchmarks.
  - Through democratization, fairness and innovation is fostered because knowledge and evaluation methods are disseminated.

8 CONCLUSION

Overall, this comprehensive paper has explored the motivations and pathways for creating a holistic benchmark carpentry effort, paying specific attention to aspects that can democratize AI benchmarks. This was achieved by (a) providing standardized and formal definitions of benchmarks, and (b) identifying a representative set of benchmarks related to AI activities. Finally, we propose an AI Benchmark Carpentry curriculum that integrates the various topics discussed into a structured learning activities to empower practitioners with reproducible coding practices, experiment-management skills, and an ethical lens on benchmarking. By embedding FAIR principles, bias-mitigation techniques, and performance-tuning modules, the curriculum offers a scalable pathway for communities—from academic labs to industry R&D—to build, share, and improve benchmarks in a collaborative, transparent manner.

Together, these activities foster democratization of AI benchmarks and can be utilized to grow the community and the understanding on how benchmarks may effect an individual activity or even community. While deploying such activities, we hope to grow community awareness and overcome the lack of well defined activities to educate the community in this regard. While fostering these activities we also address the need for more easily develop dynamic and adaptable benchmarks.

9 NOMENCLATURE

9.1 Resource Identification Initiative

To take part in the Resource Identification Initiative, please use the corresponding catalog number and RRID in your current manuscript. For more information about the project and for steps on how to search for an RRID, please click here.

9.2 Life Science Identifiers

Life Science Identifiers (LSIDs) for ZOOBANK registered names or nomenclatural acts should be listed in the manuscript before the keywords. For more information on LSIDs please see Inclusion of Zoological Nomenclature section of the guidelines.

10 ADDITIONAL REQUIREMENTS

For additional requirements for specific article types and further information please refer to Author Guidelines.

CONFLICT OF INTEREST STATEMENT

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

AUTHOR CONTRIBUTIONS

- **Gregor von Laszewski** is the lead author of the paper. He identified first that efforts in benchmark carpentry and democratization are needed. He has lead the organization of this paper in the MLCommons Science Working Group. He also created the initial version of [152] which is related and relevant to this effort.
- **Piotr Luszczek** has contributed to integration of many decades of experiences from designing, implementing, running, and collecting results from HPC benchmarks.
- **Wesley Brewer** has contributed to the simulation section.
- **Jeyan Thiyagalingam** has worked on the GPU benchmarking section. Reviewed the paper and made corrections.
- **Juri Papay** has worked on the GPU benchmarking section. Updated the GPU HW details of MLCommon benchmarks.

- **Geoffrey C. Fox** is leading the MLCommons Science Working group and has contributed to many of the ideas. The experiences and discussions with Gregor von Laszewski around improvements to the earthquake benchmark have significantly contributed to this effort. The educational effort of using the earthquake benchmark with a number of students motivated this effort.
- **Armstrong Foundjem** has provided an early version and led the Energy section, and has contributed to the paper writing and overall improvement.
- **Gregg Barrett** has participated in discussions as part of the working group meetings and contributed to an early version of this paper.
- **Murali Emani** has participated in discussions as part of the working group meetings and improved the article.
- **Shirley V. Moore** has written text for the Profiling and Performance Analysis Section.
- **Vijay Janapa Reddi** has participated in discussions as part of the working group meetings and improved the article.
- **Matthew D. Sinclair, Shivaram Venkataraman and Rutwik Jain** participated in discussions as part of the working group meetings and wrote the variability section of the article. Sinclair also wrote the simulation section of the paper and helped improve the paper in other sections.
- **Christine Kirkpatrick** has worked on conceptualizing the ideas and discussion, and helping with the Carpentries background section.
- **Kartik Mathur** Has worked on improving an early version of the Energy section.
- **Victor Lu** Has participated in writing the paper.
- **Tianhao Li** has participated in discussions as part of the working group meetings and participated in identification of limitations of current benchmarks.
- **Sebastian Lobentanzer** Has participated in the discussions in the working group and has contributed to the abstract, intro, definitions, formalization, and benchmark sections with content and editing.
- **Sujata Goswami** has worked on the MLCommons benchmark details in Table 1.
- **Abdulkareem Alsudais** has reviewed the motivation to AI Benchmark Carpentry and contributed to the writing of this paper.
- **Kongtao Chen** has worked on the monitoring sections, related benchmarks, and participated in discussions as part of the working group meetings.
- **Tejinder Singh** has edited and improved AI hardware benchmarking and infrastructure sections and provided new KPIs for AI hardware benchmarking.
- **Kirsten Morehouse knmorehouse@gmail.com** has participated in discussions as part of the working group meetings. Morehouse also reviewed the paper and made improvements.
- **Marco Colombo, Benjamin Hawks, and Nhan Tran** have worked on the benchmark ontology and Table 2.
- **Khojasteh Z. Mirza** has participated in discussions as part of the working group meetings and worked an a very early version of the energy section.
- **Renato Umeton** revised the manuscript for consistency and coherence.

- **Sasidhar Kunapuli and Gavin Farrell gavinmichael.farrell@phd.unipd.it** have participated in discussions as part of the working group meetings.
- **Gary Mazzaferro** has participated in discussions as part of the working group meetings surrounding benchmark definitions and applicability.

FUNDING

Details of all funding sources should be provided, including grant numbers if applicable. Please ensure to add all necessary funding information, as after publication this is no longer possible.

The work was in part sponsored by NSF Grant #2346173 and # 2303700, POSE: Phase II: MLCommons Research for Science: Enabling Open-Source Ecosystems for Scientific Foundation Models by Community Standards and Benchmarks

The portion of this work done at UW-Madison is supported in part by NSF grant CNS-2312688 and by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under Award Number DE-SC-0026036.

This manuscript has been in part authored by FermiForward Discovery Group, LLC under Contract No. 89243024CSC000002 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics. Fermilab Report Number FERMILAB-PUB-25-0835-CSAID.

This work was supported by DOE ASCR Microelectronics Science Research Center Projects, BIA. This material is based upon work supported by the U.S. Department of Energy, Office of Science, under contract number DE-AC02-06CH11357.

This research was in part sponsored in part by and used resources of the Oak Ridge Leadership Computing Facility (OLCF), which is a DOE Office of Science User Facility at the Oak Ridge National Laboratory (ORNL) supported by the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

Shirley Moore's work on this paper was supported by the Department of Energy Office of Science under award #DE-SC0024352.

Kirkpatrick's work was made possible through the National Science Foundation award #2226453.

This research was funded in part by and used resources at the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

Research was sponsored by the Department of the Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Department of the Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

ACKNOWLEDGMENTS

We have used at one point “ChatGPT” to improve upon the grammar of selected sections with the question: “Improve the grammar of ...”. However, we stopped that practice early on due to wrong corrections, and have used Grammarly throughout the paper.

SUPPLEMENTAL DATA

Supplementary Material should be uploaded separately on submission, if there are Supplementary Figures, please include the caption in the same file as the figure. LaTeX Supplementary Material templates can be found in the Frontiers LaTeX folder.

DATA AVAILABILITY STATEMENT

The datasets [GENERATED/ANALYZED] for this study can be found in the [NAME OF REPOSITORY] [LINK].

REFERENCES

[1] DOE, “Trillion parameter consortium,” Aug. 2023, [Online; accessed 2025-11-30]. [Online]. Available: https://tpc.dev/

[2] R. Stevens, “Argonne’s "AuroraGPT" Project,” *Trillion Parameter Consortium Seminar*, 2023.

[3] Wikipedia, “Benchmark (computing),” 6 2005, [Online; accessed 2025-09-23]. [Online]. Available: https://en.wikipedia.org/wiki/Benchmark_%28computing%29

[4] J. J. Dongarra, “Performance of various computers using standard linear equations software,” University of Tennessee, Knoxville / Oak Ridge National Laboratory, Tech. Rep. Technical Report CS-89-85, 1989. [Online]. Available: http://www.netlib.org/benchmark/performance.ps

[5] J. J. Dongarra, M. A. Heroux, and P. Luszczek, “High-performance conjugate-gradient benchmark: A new metric for ranking high-performance computing systems,” *International Journal of High Performance Computing Applications*, vol. 30, no. 1, pp. 3–8, 2016. [Online]. Available: https://doi.org/10.1177/1094342015593158

[6] Google Cloud Platform and contributors, “Perfkitbenchmarker,” GitHub, 2025. [Online]. Available: https://github.com/GoogleCloudPlatform/PerfKitBenchmarker

[7] I. S. Committee, “Io500: A benchmarking suite for hpc storage i/o performance,” Web Page, 2025. [Online]. Available: https://io500.org

[8] G. Wilson, “Software carpentry: lessons learned,” *F1000Research*, vol. 3, 2014. [Online]. Available: https://doi.org/10.12688/f1000research.3-62.v2

[9] Software Carpentry, “Software carpentry,” https://software-carpentry.org/, 2024, accessed: 2025-05-28.

[10] The Carpentries, “Data carpentry,” https://datacarpentry.org, 2025, accessed: 2025-10-23.

[11] A. Reid, T. Keller, A. O’Cais, A. A. Rasel, W. Purwanto, J. Herriman, B. Muite, and M.-A. Hermanns, “Hpc carpentry: Recent progress and incubation toward an official carpentries lesson program,” *Journal of Computational Science*, vol. 16, no. 1, 2025.

[12] The Carpentries / HPC Carpentry community, “Hpc carpentry,” https://hpc-carpentry.org, 2025, accessed: 2025-10-23.

[13] G. von Laszewski, J. P. Fleischer, R. Knuuti, G. C. Fox, J. Kolessar, T. S. Butler, and J. Fox, “Opportunities for enhancing mlcommons efforts while leveraging insights from educational mlcommons earthquake benchmarks efforts,” *Frontiers in High Performance Computing*, vol. 1, no. 1233877, p. 31, October 2023. [Online]. Available: https://www.frontiersin.org/journals/high-performance-computing/articles/10.3389/fhpcp.2023.1233877

[14] J. J. Dongarra, P. Luszczek, and A. Petitet, “The LINPACK benchmark: Past, present, and future,” *Concurrency and Computation: Practice and Experience*, vol. 15, no. 9, pp. 803–820, August 10 2003, iSSN 1532-0634.

[15] J. J. Dongarra, J. Bunch, C. Moler, and G. W. Stewart, *LINPACK User’s Guide*. Philadelphia, PA, USA: Society of Industrial and Applied Mathematics, 1979.

[16] W.-C. Feng, R. Ge, and K. W. Cameron, “Power and energy profiling of scientific applications on distributed systems,” in *19th IEEE International Parallel and Distributed Processing Symposium (IPDPS 05)*, Denver, CO, USA, 2005.

[17] K. W. Cameron, R. Ge, and X. Feng, “High-performance, power-aware, distributed computing for scientific applications,” *IEEE Computer*, vol. 38, no. 11, pp. 40–47, 2005.

[18] SPEC, “The SPEC power benchmark,” 2008. [Online]. Available: www.spec.org/power_ssj2008/

[19] “Specpower_ssj2008,” https://spec.org/power_ssj2008/, 2025, watts per transaction and operations per Watt for enterprise servers.

[20] W.-c. Feng and K. Cameron, “The green500 list: Encouraging sustainable supercomputing,” *Computer*, vol. 40, no. 12, p. 50–55, dec 2007. [Online]. Available: https://doi.org/10.1109/MC.2007.445

[21] A. Abdelfattah, H. Anzt, E. G. Boman, E. Carson, T. Cojean, J. Dongarra, A. Fox, M. Gates, N. J. Higham, X. S. Li, J. Loe, P. Luszczek, S. Pranesh, S. Rajamanickam, T. Ribizel, B. F. Smith, K. Swirydowicz, S. Thomas, S. Tomov, Y. M. Tsai, and U. M. Yang, “A survey of numerical linear algebra methods utilizing mixed-precision arithmetic,” *The International Journal of High Performance Computing Applications*, vol. 35, no. 4, pp. 344–369, 2021. [Online]. Available: https://doi.org/10.1177/10943420211003313

[22] Cornell University, “arxiv.org e-print archive,” Oct. 2025, [Online; accessed 2025-10-01]. [Online]. Available: https://arxiv.org/

[23] “Google scholar,” Oct. 2025, [Online; accessed 2025-10-01]. [Online]. Available: https://scholar.google.com/

[24] G. von Laszewski, B. Hawks, M. Colombo, R. Shiraishi, A. Krishnan, N. Tran, and G. C. Fox, “Mlcommons science working group ai benchmarks collection,” GitHub, Jun. 2025, online Collection: =https://mlcommons-science.github.io/benchmark/. [Online]. Available: https://mlcommons-science.github.io/benchmark/benchmarks.pdf

[25] B. Hawks, G. von Laszewski, M. D. Sinclair, M. Colombo, S. Venkataraman, R. Jain, Y. Jiang, N. Tran, and G. Fox, “An MLCommons Scientific Benchmarks Ontology,” arXiv, 2025. [Online]. Available: https://arxiv.org/abs/2511.05614

[26] “Machine learning innovation to benefit everyone,” *Web page*, Apr. 2023. [Online]. Available: https://mlcommons.org/

[27] G. von Laszewski, N. Tran, and others, “mlcommons-science/benchmark,” Oct. 2025, [Online; accessed 2025-10-01]. [Online]. Available: https://github.com/mlcommons-science/benchmark

[28] G. von Laszewski, J. Fleischer, G. C. Fox, J. Papay, S. Jackson, and J. Thiyagalingam, “Templated hybrid reusable computational analytics workflow management with cloudmesh,

applied to the deep learning mlcommons cloudmask application," in *2023 IEEE 19th International Conference on e-Science (e-Science)*, 2023, pp. 1–6.

[29] S. Che, B. M. Beckmann, S. K. Reinhardt, and K. Skadron, "Pannotia: Understanding Irregular GPGPU Graph Applications," in *IEEE International Symposium on Workload Characterization*, ser. IISWC, 9 2013, pp. 185–195.

[30] Y. Wang, Y. Pan, A. Davidson, Y. Wu, C. Yang, L. Wang, M. Osama, C. Yuan, W. Liu, A. T. Riffel, and J. D. Owens, "Gunrock: GPU Graph Analytics," *ACM Trans. Parallel Comput.*, vol. 4, no. 1, Aug 2017. [Online]. Available: https://doi.org/10.1145/3108140

[31] Lawrence Livermore National Labs, "CORAL-2 Benchmarks," https://asc.llnl.gov/coral-2-benchmarks, 2020.

[32] Oak Ridge National Labs, "OLCF-6 Benchmarks," https://www.olcf.ornl.gov/draft-olcf-6-technical-requirements/benchmarks/, 2024.

[33] J. Kim, A. D. Baczewski, T. D. Beaudet, A. Benali, M. C. Bennett, M. A. Berrill, N. S. Blunt, E. J. L. Borda, M. Casula, D. M. Ceperley *et al.*, "QMCPACK: An Open Source ab initio Quantum Monte Carlo Package for the Electronic Structure of Atoms, Molecules and Solids," *Journal of Physics: Condensed Matter*, vol. 30, no. 19, p. 195901, 2018.

[34] X. Wu, V. Taylor, J. M. Wozniak, R. Stevens, T. Brettin, and F. Xia, "Performance, Energy, and Scalability Analysis and Improvement of Parallel Cancer Deep Learning CANDLE Benchmarks," in *Proceedings of the 48th International Conference on Parallel Processing*, ser. ICPP. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3337821.3337905

[35] C. R. Banbury, V. J. Reddi, P. Torelli, J. Holleman, N. Jeffries, C. Király, P. Montino, D. Kanter, S. Ahmed, D. Pau, U. Thakker, A. Torrini, P. Warden, J. Cordaro, G. D. Guglielmo, J. M. Duarte, S. Gibellini, V. Parekh, H. Tran, N. Tran, W. Niu, and X. Xu, "Mlperf tiny benchmark," *CoRR*, vol. abs/2106.07597, 2021. [Online]. Available: https://arxiv.org/abs/2106.07597

[36] T. Baruah, K. Shivdikar, S. Dong, Y. Sun, S. A. Mojumder, K. Jung, J. L. Abellán, Y. Ukidave, A. Joshi, J. Kim, and D. Kaeli, "GNNMark: A Benchmark Suite to Characterize Graph Neural Network Training on GPUs," in *IEEE International Symposium on Performance Analysis of Systems and Software*, ser. ISPASS, 2021, pp. 13–23.

[37] S. Dong and D. Kaeli, "DNNMark: A Deep Neural Network Benchmark Suite for GPUs," in *Proceedings of the General Purpose GPUs*, ser. GPGPU. New York, NY, USA: ACM, 2017, pp. 63–72. [Online]. Available: http://doi.acm.org/10.1145/3038228.3038239

[38] S. Narang and G. Diamos, "An update to DeepBench with a focus on deep learning inference," https://svail.github.io/DeepBench-update/, 2017.

[39] P. Mattson, C. Cheng, C. Coleman, G. Diamos, P. Micikevicius, D. A. Patterson, H. Tang, G. Wei, P. Bailis, V. Bittorf, D. Brooks, D. Chen, D. Dutta, U. Gupta, K. M. Hazelwood, A. Hock, X. Huang, B. Jia, D. Kang, D. Kanter, N. Kumar, J. Liao, G. Ma, D. Narayanan, T. Oguntebi, G. Pekhimenko, L. Pentecost, V. J. Reddi, T. Robie, T. S. John, C. Wu, L. Xu, C. Young, and M. Zaharia, "MLPerf Training Benchmark," *CoRR*, vol. abs/1910.01500, 2019. [Online]. Available: http://arxiv.org/abs/1910.01500

[40] P. Mattson, V. J. Reddi, C. Cheng, C. Coleman, G. Diamos, D. Kanter, P. Micikevicius, D. Patterson, G. Schmuelling, H. Tang *et al.*, "MLPerf: An Industry Standard Benchmark Suite for Machine Learning Performance," *IEEE Micro*, vol. 40, no. 2, pp. 8–16, 2020.

[41] V. J. Reddi, C. Cheng, D. Kanter, P. Mattson, G. Schmuelling, C.-J. Wu, B. Anderson, M. Breughe, M. Charlebois, W. Chou, R. Chukka, C. Coleman, S. Davis, P. Deng, G. Diamos, J. Duke, D. Fick, J. S. Gardner, I. Hubara, S. Idgunji, T. B. Jablin, J. Jiao, T. S. John, P. Kanwar, D. Lee, J. Liao, A. Lokhmotov, F. Massa, P. Meng, P. Micikevicius, C. Osborne, G. Pekhimenko, A. T. R. Rajan, D. Sequeira, A. Sirasao, F. Sun, H. Tang, M. Thomson, F. Wei, E. Wu, L. Xu, K. Yamada, B. Yu, G. Yuan, A. Zhong, P. Zhang, and Y. Zhou, "MLPerf Inference Benchmark," in *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture*, ser. ISCA, 2020, pp. 446–459.

[42] V. J. Reddi, C. Cheng, D. Kanter, P. Mattson, G. Schmuelling, and C.-J. Wu, "The Vision Behind MLPerf: Understanding AI Inference Performance," *IEEE Micro*, vol. 41, no. 3, pp. 10–18, 2021.

[43] Z. Fan and E. Ma, "Predicting orientation-dependent plastic susceptibility from static structure in amorphous solids via deep learning," *Nature communications*, vol. 12, no. 1, pp. 1–13, 2021.

[44] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko *et al.*, "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021.

[45] J. Kates-Harbeck, A. Svyatkovskiy, and W. Tang, "Predicting disruptive instabilities in controlled fusion plasmas through deep learning," *Nature*, vol. 568, no. 7753, pp. 526–531, 2019.

[46] J. Thiyagalingam, M. Shankar, G. Fox, and T. Hey, "Scientific Machine Learning Benchmarks," *Nature Reviews Physics*, vol. 4, no. 6, pp. 413–420, 2022.

[47] J. Thiyagalingam, G. von Laszewski, J. Yin, M. Emani, J. Papay, G. Barrett, P. Luszczek, A. Tsaris, C. Kirkpatrick, F. Wang, T. Gibbs, V. Vishwanath, M. Shankar, G. Fox, and T. Hey, "Ai benchmarking for science: Efforts from the mlcommons science working group," in *High Performance Computing. ISC High Performance 2022 International Workshops*, H. Anzt, A. Bienz, P. Luszczek, and M. Baboulin, Eds. Cham: Springer International Publishing, 2022, pp. 47–64.

[48] H. Wang, L. Zhang, J. Han, and W. E, "DeePMD-kit: A deep learning package for many-body potential energy representation and molecular dynamics," *Computer Physics Communications*, vol. 228, pp. 178–184, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010465518300882

[49] J. Zeng, D. Zhang, D. Lu, P. Mo, Z. Li, Y. Chen, M. Rynik, L. Huang, Z. Li, S. Shi, Y. Wang, H. Ye, P. Tuo, J. Yang, Y. Ding, Y. Li, D. Tisi, Q. Zeng, H. Bao, Y. Xia, J. Huang, K. Muraoka, Y. Wang, J. Chang, F. Yuan, S. L. Bore, C. Cai, Y. Lin, B. Wang, J. Xu, J.-X. Zhu, C. Luo, Y. Zhang, R. E. A. Goodall, W. Liang, A. K. Singh, S. Yao, J. Zhang, R. Wentzcovitch, J. Han, J. Liu, W. Jia, D. M. York, W. E, R. Car, L. Zhang, and H. Wang, "DeePMD-kit v2: A software package for deep potential models," *The Journal of Chemical Physics*, vol. 159, no. 5, p. 054801, 08 2023. [Online]. Available: https://doi.org/10.1063/5.0155600

[50] G. Derevyanko, G. Lamoureux, C. Outeiral, T. Oda, F. Fuchs, S. P. Mahajan, J. Moult, J. Haas, P. Maragakis, T. Ruzmetov, and M. AlQuraishi, "OpenFold2: Replicating AlphaFold2 in the Dark," https://lupoglaz.github.io/OpenFold2/, 2023.

[51] B. Acun, A. Langer, E. Meneses, H. Menon, O. Sarood, E. Totoni, and L. V. Kalé, "Power, Reliability, and Performance: One System to Rule them All," *Computer*, vol. 49, no. 10, pp. 30–37, 2016.

[52] D. Chasapis, M. Casas, M. Moretó, M. Schulz, E. Ayguadé, J. Labarta, and M. Valero, "Runtime-Guided Mitigation of Manufacturing Variability in Power-Constrained Multi-Socket NUMA Nodes," in *Proceedings of the 2016 International Conference on Supercomputing*, ser. ICS '16, 2016.

[53] D. Chasapis, M. Moretó, M. Schulz, B. Rountree, M. Valero, and M. Casas, "Power Efficient Job Scheduling by Predicting the Impact of Processor Manufacturing Variability," in *Proceedings of the ACM International Conference on Supercomputing*, ser. ICS '19, 2019, p. 296–307. [Online]. Available: https://doi.org/10.1145/3330345.3330372

[54] Y. Inadomi, T. Patki, K. Inoue, M. Aoyagi, B. Rountree, M. Schulz, D. Lowenthal, Y. Wada, K. Fukazawa, M. Ueda, M. Kondo, and I. Miyoshi, "Analyzing and Mitigating the Impact of Manufacturing Variability in Power-Constrained Supercomputing," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC, 2015. [Online]. Available: https://doi.org/10.1145/2807591.2807638

[55] T. Patel, A. Wagenhäuser, C. Eibel, T. Hönig, T. Zeiser, and D. Tiwari, "What does Power Consumption Behavior of HPC Jobs Reveal? : Demystifying, Quantifying, and Predicting Power Consumption Characteristics," in *IEEE International Parallel and Distributed Processing Symposium*, ser. IPDPS, 2020, pp. 799–809.

[56] D. Skinner and W. Kramer, "Understanding the causes of performance variability in HPC workloads," in *Proceedings of the IEEE Workload Characterization Symposium*, ser. IISWC, 2005, pp. 137–149.

[57] T. Scogland, J. Azose, D. Rohr, S. Rivoire, N. Bates, and D. Hackenberg, "Node Variability in Large-Scale Power Measurements: Perspectives from the Green500, Top500 and EEHPCWG," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '15. New York, NY, USA: Association for Computing Machinery, 2015. [Online]. Available: https://doi.org/10.1145/2807591.2807653

[58] N. DeBardeleben, S. Blanchard, L. Monroe, P. Romero, D. Grunau, C. Idler, and C. Wright, "GPU Behavior on a Large HPC Cluster," in *Euro-Par 2013: Parallel Processing Workshops - BigDataCloud, DIHC, FedICI, HeteroPar, HiBB, LSDVE, MHPC, OMHI, PADABS, PROPER, Resilience, ROME, and UCHPC 2013, Aachen, Germany, August 26-27, 2013. Revised Selected Papers*, ser. Lecture Notes in Computer Science, D. an Mey, M. Alexander, P. Bientinesi, M. Cannataro, C. Clauss, A. Costan, G. Kecskemeti, C. Morin, L. Ricci, J. Sahuquillo, M. Schulz, V. Scarano, S. L. Scott, and J. Weidendorfer, Eds., vol. 8374. Springer, 2013, pp. 680–689. [Online]. Available: https://doi.org/10.1007/978-3-642-54420-0_66

[59] D. De Sensi, T. De Matteis, K. Taranov, S. Di Girolamo, T. Rahn, and T. Hoefler, "Noise in the Clouds: Influence of Network Performance Variability on Application Scalability," vol. 6, no. 3, Dec. 2022. [Online]. Available: https://doi.org/10.1145/3570609

[60] F. Fraternali, A. Bartolini, C. Cavazzoni, and L. Benini, "Quantifying the Impact of Variability and Heterogeneity on the Energy Efficiency for a Next-Generation Ultra-Green Supercomputer," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 7, pp. 1575–1588, 2018.

[61] E. Sencan, D. Kulkarni, A. Coskun, and K. Konate, "Analyzing GPU Utilization in HPC Workloads: Insights from Large-Scale Systems," in *Practice and Experience in Advanced Research Computing 2025: The Power of Collaboration*, ser. PEARC. New York, NY, USA: Association for Computing Machinery, 2025. [Online]. Available: https://doi.org/10.1145/3708035.3736010

[62] P. Sinha, A. Guliani, R. Jain, B. Tran, M. D. Sinclair, and S. Venkataraman, "Not All GPUs Are Created Equal: Characterizing Variability in Large-Scale, Accelerator-Rich Systems," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, ser. SC, 2022, pp. 1–15.

[63] B. Topcu, D. Karabacak, and I. Oz, "Demystifying Power and Performance Variations in GPU Systems through Microarchitectural Analysis," *Computer Science and Information Systems*, vol. 22, no. 2, pp. 533–561, 2025.

[64] X. You, Z. Xuan, H. Yang, Z. Luan, Y. Liu, and D. Qian, "GVARP: Detecting Performance Variance on Large-Scale Heterogeneous System," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis*, ser. SC, 2024.

[65] Z. Zhong, S. Sultanov, M. Papka, and Z. Lan, "Minimizing Power Waste in Heterogenous Computing via Adaptive Uncore Scaling," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, ser. SC, 2025.

[66] K. Heyman, "DRAM Thermal Issues Reach Crisis Point," https://semiengineering.com/dram-thermal-issues-reach-crisis-point/, 2022.

[67] D. Stanzione, J. West, R. T. Evans, T. Minyard, O. Ghattas, and D. K. Panda, "Frontera: The evolution of leadership computing at the National Science Foundation," in *Practice and Experience in Advanced Research Computing*, 2020, pp. 106–111.

[68] TACC, "Texas Advanced Computing Center," https://www.tacc.utexas.edu/, 2021.

[69] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS-W*, 2017.

[70] R. Jain, B. Tran, K. Chen, M. D. Sinclair, and S. Venkataraman, "PAL: A Variability-Aware Policy for Scheduling ML Workloads in GPU Clusters," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis*, ser. SC, November 2024.

[71] J. Guerreiro, A. Ilic, N. Roma, and P. Tomás, "DVFS-aware application classification to improve GPGPUs energy efficiency," *Parallel Computing*, vol. 83, pp. 93–117, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167819118300243

[72] Science Feedback, "Training and Using ChatGPT Uses a Lot of Energy, but Exact Numbers Are Tricky to
Pin Down Without Data from OpenAI," 2024. [Online]. Available: https://science.feedback.org/training-and-using-chatgpt-uses-a-lot-of-energy-but-exact-numbers-are-tricky-to-pin-down-without-dat

[73] 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. [Online]. Available: https://arxiv.org/abs/2001.08361

[74] U. E. I. Administration, "Average price of electricity to ultimate customers," https://www.eia.gov/electricity/monthly/, 2025.

[75] U.S. Energy Information Administration, "Average annual electricity consumption for u.s. residential customers," Available at https://www.eia.gov/, 2024.

[76] World Economic Forum, “Ai and energy: Will ai help reduce emissions or increase power demand? here’s what to know,” *NA*, Jul. 2024, (Accessed on 09/07/2025). [Online]. Available: https://www.weforum.org/stories/2024/07/generative-ai-energy-emissions/

[77] D. Patterson, J. Gonzalez, Q. V. Le, C. Liang, L. Munguia, D. Rothchild, D. R. So, M. Texier, and J. Dean, “The Carbon Footprint of Large Neural Network Training,” *arXiv preprint arXiv:2104.10350*, 2021. [Online]. Available: https://arxiv.org/abs/2104.10350

[78] N. Jegham, M. Abdelatti, C. Y. Koh, L. Elmoubarki, and A. Hendawi, “How hungry is ai? benchmarking energy, water, and carbon footprint of llm inference,” 2025. [Online]. Available: https://arxiv.org/abs/2505.09598

[79] Baeldung Editors, “How Much Energy Does ChatGPT Use?” 2023. [Online]. Available: https://www.baeldung.com/cs/chatgpt-large-language-models-power-consumption

[80] Data Science on Medium, “The Carbon Footprint of GPT-4,” 2023. [Online]. Available: https://medium.com/data-science/the-carbon-footprint-of-gpt-4-d6c676eb21ae

[81] Extreme Networks, “Confronting AI’s Growing Energy Appetite: Part 1,” 2023. [Online]. Available: https://www.extremenetworks.com/resources/blogs/confronting-ai-growing-energy-appetite-part-1

[82] Epoch AI, “Why GPT-5 Used Less Training Compute Than GPT-4.5 (But GPT-6 Probably Won’t),” 2024. [Online]. Available: https://epoch.ai/gradient-updates/why-gpt5-used-less-training-compute-than-gpt45-but-gpt6-probably-wont

[83] H. Editors, “AI’s Dirty Secret: The Energy Cost of Training the Next GPT-5,” 2024. [Online]. Available: https://hackernoon.com/ais-dirty-secret-the-energy-cost-of-training-the-next-gpt-5

[84] A. S. Bland, J. H. Rogers II, R. A. Kendall, D. B. Kothe, and G. M. Shipman, “Jaguar: The World’s Most Powerful Computer,” in *35th Cray User Group Meeting*, ser. CUG, 5 2009.

[85] B. Bland, “Titan - Early experience with the Titan system at Oak Ridge National Laboratory,” in *SC Companion: High Performance Computing, Networking Storage and Analysis*, 2012, pp. 2189–2211.

[86] D. E. Womble, M. Shankar, W. Joubert, J. T. Johnston, J. C. Wells, and J. A. Nichols, “Early Experiences on Summit: Data Analytics and AI Applications,” *IBM Journal of Research and Development*, vol. 63, no. 6, pp. 2:1–2:9, 2019.

[87] S. Atchley, C. Zimmer, J. R. Lange, D. E. Bernholdt, V. G. Melesse Vergara, T. Beck, M. J. Brim, R. Budiardja, S. Chandrasekaran, M. Eisenbach, T. Evans, M. Ezell, N. Frontiere, A. Georgiadou, J. Glenski, P. Grete, S. Hamilton, J. Holmen, A. Huebl, D. Jacobson, W. Joubert, K. McMahon, E. Merzari, S. G. Moore, A. Myers, S. Nichols, S. Oral, T. Papatheodore, D. Perez, D. M. Rogers, E. Schneider, J.-L. Vay, and P. Yeung, “Frontier: Exploring Exascale The System Architecture of the First Exascale Supercomputer,” in *International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC, 2023, pp. 1–16.

[88] T. R. W. Scogland, B. Subramaniam, and W. Feng, “Emerging Trends on the Evolving Green500: Year Three,” in *Proc. IEEE IPDPS Workshops*, 2011, pp. 889–895.

[89] A. Tschand, A. T. R. Rajan, S. Idgunji, A. Ghosh, J. Holleman, C. Kiraly, P. Ambalkar, R. Borkar, R. Chukka, T. Cockrell, O. Curtis, G. Fursin, M. Hodak, H. Kassa, A. Lokhmotov, D. Miskovic, Y. Pan, M. P. Manmathan, L. Raymond, T. S. John, A. Suresh, R. Taubitz, S. Zhan, S. Wasson, D. Kanter, and V. J. Reddi, “MLPerf Power: Benchmarking the Energy Efficiency of Machine Learning Systems from μWatts to MWatts for Sustainable AI,” in *IEEE*

International Symposium on High Performance Computer Architecture, ser. HPCA, 2025, pp. 1201–1216.

[90] "Spec sert^2," https://spec.org/sert2/, 2025, server Efficiency Rating Tool with calibrated energy measurements.

[91] "Tpc-energy," https://www.tpc.org/, 2025, energy add-on kit for TPC database benchmarks.

[92] "Joulesort benchmark," https://sortbenchmark.org/, 2025, records sorted per Joule; storage I/O energy efficiency.

[93] "Hpcg-power," https://hpcg-benchmark.org/, 2025, energy efficiency (GFLOPS/W) for the High Performance Conjugate Gradient benchmark.

[94] "Hpl-mxp (hpl-ai)," https://top500.org/news/hpl-ai-benchmark/, 2025, mixed-precision LINPACK benchmark with GFLOPS/W metric.

[95] "Mlperf power: Training and inference," https://mlcommons.org/en/power/, 2025, joules, average Watts, Joules per sample/epoch for ML workloads.

[96] "Mlperf tiny: Energy mode," https://mlcommons.org/en/tiny/, 2025, microjoules per inference on micro-controllers.

[97] "Coremark-pro power," https://www.eembc.org/coremarkpro/, 2025, iterations per second per Watt for embedded/SoC devices.

[98] "Ul procyon ai inference power test," https://benchmarks.ul.com/procyon, 2025, images per Watt and fps/W on desktop and mobile devices.

[99] "Candle power study (sc19)," https://doi.org/10.1145/3337821.3337924, 2025, deep learning cancer benchmark with Joules/epoch & GFLOPS/W metrics.

[100] "Lulesh/minife energy benchmark," https://doi.org/10.1109/ISPA-BDCloud-SocialCom-SustainCom55337.2022.00045, 2025, energy/Joules per iteration for proxy-apps (Gerofi et al., 2022).

[101] "Exasmr power benchmark," https://doi.org/10.1016/j.jpdc.2021.05.001, 2025, energy vs accuracy trade-off for neutron transport mini-app.

[102] "Ee-hpc-wg energy benchmark (draft)," https://eehpcwg.llnl.gov/, 2025, community draft specification for node & job energy benchmarking.

[103] "Hpc-ai500 energy track (planned)," https://www.hpc-ai.org/, 2025, upcoming GFLOPS/W extension to HPC-AI500 mixed AI/HPC benchmark.

[104] "Parsec-3.1 energy extension," https://parsec.cs.gatech.edu/, 2025, research prototype adding power metrics to PARSEC benchmark suite.

[105] Prabhat *et al.*, "Scaling CosmoFlow to 15,000 GPUs and achieving 43 pflops," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC19)*, 2019, includes CosmoFlow-Power joules/epoch data.

[106] K. Heitmann *et al.*, "The hacc framework: Energy and performance characterization," *Computing in Science & Engineering*, 2020, adds HACC Energy Add-on joules/particle metric.

[107] T. Kurth *et al.*, "Exascale deep learning for climate analytics," in *International Conference for High Performance Computing (SC20)*, 2020, deepCAM-Energy joules/epoch results.

[108] N. Wedi *et al.*, "Openifs energy benchmark report," ECMWF, Tech. Rep., 2023, kWh per model-day for full weather physics. [Online]. Available: https://www.ecmwf.int/en/publications/openifs/energy-benchmark

[109] S. Páll *et al.*, "Gromacs-ee: Energy-efficient molecular dynamics on gpus," in *GPU Technology Conference (GTC)*, 2024, introduces Joules/ns metric. [Online]. Available: https://developer.nvidia.com/gtc

[110] A. Rodriguez *et al.*, “Energy delay product optimization of namd on summit,” *Journal of Computational Chemistry*, 2019, energy-Delay Product results for ApoA1.

[111] P. Giannozzi *et al.*, “Energy-aware quantum espresso: Joules per scf step,” in *International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computing Systems (PMBS)*, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9955431

[112] G. Kresse *et al.*, “Vasp power harness: Energy profiling of dft md,” Vienna University of Technology, Tech. Rep., 2023. [Online]. Available: https://vasp.at/energy-harness

[113] R. Jain *et al.*, “Characterizing energy consumption of openfoam on modern hpc systems,” in *Workshop on Energy Efficient Supercomputing*, 2021.

[114] T. Farr *et al.*, “Insar-ai: Power characterization of satellite image unwrapping,” *IEEE Journal of Selected Topics in Applied Earth Observations*, 2024, joules per satellite scene.

[115] G. Fox *et al.*, “H3d: Hydrology 3d energy benchmark,” in *International Conference on Computational Science (ICCS)*, 2023, joules per timestep metric. [Online]. Available: https://iccs2023.org

[116] “Spec ptdaemon / sert energy for hpc,” https://spec.org/ptdaemon/, 2025, calibrated power logging used with SPEC benchmarks on HPC systems.

[117] “Scaphandre,” https://github.com/hubblo-org/scaphandre, 2025, process & node power telemetry agent for Linux clusters (Watts, kWh).

[118] “Kepler: Kubernetes-based energy profiler,” https://github.com/sustainable-computing-io/kepler, 2025, watts and Joules per container/pod using eBPF/RAPL.

[119] “Codecarbon,” https://codecarbon.io/, 2025, process-level kWh and kg CO2e estimation library.

[120] “Carbontracker,” https://github.com/lfwa/carbontracker, 2025, energy and CO2 prediction for deep-learning training.

[121] “Powerpack / mont-blanc,” https://gitlab.bsc.es/mont-blanc/PowerPACK, 2025, energy & power profiling toolkit for MPI/OpenMP mini-apps (Joules, Watts).

[122] “Cray pat energy counters,” https://support.hpe.com/hpesc/public/docDisplay?docId=a00111513en_us, 2025, integrated energy-per-function profiling in HPE/Cray Performance Analysis Tool.

[123] “Ibm powerapi (pmlib),” https://github.com/IBM/powerapi, 2025, system & per-process kWh reporting on Power-based supercomputers.

[124] “Nvidia dcgm energy,” https://developer.nvidia.com/dcgm, 2025, gPU Joules & Watts via Data Center GPU Manager; attachable to HPC benchmarks.

[125] “Intel vtune power analysis,” https://www.intel.com/content/www/us/en/developer/tools/oneapi/vtune-profiler.html, 2025, package Watts & energy per function for MPI/OpenMP codes.

[126] G. von Laszewski, “Cloudmesh gpu monitor,” Feb. 2022, [Online; accessed 2025-11-26]. [Online]. Available: https://github.com/cloudmesh/cloudmesh-gpu

[127] M. Khairy, Z. Shen, T. M. Aamodt, and T. G. Rogers, “Accel-Sim: An Extensible Simulation Framework for Validated GPU Modeling,” in *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture*, ser. ISCA, 2020, pp. 473–486.

[128] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and

D. A. Wood, “The gem5 simulator,” *ACM SIGARCH Computer Architecture News*, vol. 39, no. 2, pp. 1–7, 2011.

[129] J. Lowe-Power, A. M. Ahmad, A. Akram, M. Alian, R. Amslinger, M. Andreozzi, A. Armejach, N. Asmussen, S. Bharadwaj, G. Black, G. Bloom, B. R. Bruce, D. R. Carvalho, J. Castrillon, L. Chen, N. Derumigny, S. Diestelhorst, W. Elsasser, M. Fariborz, A. Farmahini-Farahani, P. Fotouhi, R. Gambord, J. Gandhi, D. Gope, T. Grass, B. Hanindhito, A. Hansson, S. Haria, A. Harris, T. Hayes, A. Herrera, M. Horsnell, S. A. R. Jafri, R. Jagtap, H. Jang, R. Jeyapaul, T. M. Jones, M. Jung, S. Kannoth, H. Khaleghzadeh, Y. Kodama, T. Krishna, T. Marinelli, C. Menard, A. Mondelli, T. Mück, O. Naji, K. Nathella, H. Nguyen, N. Nikoleris, L. E. Olson, M. Orr, B. Pham, P. Prieto, T. Reddy, A. Roelke, M. Samani, A. Sandberg, J. Setoain, B. Shingarov, M. D. Sinclair, T. Ta, R. Thakur, G. Travaglini, M. Upton, N. Vaish, I. Vougioukas, Z. Wang, N. Wehn, C. Weis, D. A. Wood, H. Yoon, and Éder F. Zulian, “The gem5 simulator: Version 20.0+,” *CoRR*, vol. abs/2007.03152, 2020.

[130] A. F. Rodrigues, K. S. Hemmert, B. W. Barrett, C. Kersey, R. Oldfield, M. Weston, R. Risen, J. Cook, P. Rosenfeld, E. Cooper-Balis, and B. Jacob, “The Structural Simulation Toolkit,” *SIGMETRICS Perform. Eval. Rev.*, vol. 38, no. 4, p. 37–42, Mar. 2011. [Online]. Available: https://doi.org/10.1145/1964218.1964225

[131] S. Nema, R. Razdan, A. Rodrigues, K. Hemmert, G. Voskuilen, D. Adak, S. Hammond, A. Awad, and C. Hughes, “ERAS: Enabling the Integration of Real-World Intellectual Properties (IPs) in Architectural Simulators,” *Sandia National Labs Tech Report*, 9 2021. [Online]. Available: https://www.osti.gov/biblio/1854734

[132] W. Brewer, M. Maiterth, V. Kumar, R. Wojda, S. Bouknight, J. Hines, W. Shin, S. Greenwood, D. Grant, W. Williams, and F. Wang, “A digital twin framework for liquid-cooled supercomputers as demonstrated at exascale,” in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, 2024.

[133] A. Bakhoda, G. L. Yuan, W. W. L. Fung, H. Wong, and T. M. Aamodt, “Analyzing CUDA workloads using a detailed GPU simulator,” in *2009 IEEE International Symposium on Performance Analysis of Systems and Software*, ser. ISPASS, April 2009, pp. 163–174.

[134] V. Kandiah, S. Peverelle, M. Khairy, A. Manjunath, J. Pan, T. G. Rogers, T. M. Aamodt, and N. Hardavellas, “AccelWattch: A Power Modeling Framework for Modern GPUs,” in *Proceedings of the 54th IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO, October 2021.

[135] J. Lew, D. Shah, S. Pati, S. Cattell, M. Zhang, A. Sandhupatla, C. Ng, N. Goli, M. D. Sinclair, T. G. Rogers, and T. M. Aamodt, “Analyzing Machine Learning Workloads Using a Detailed GPU Simulator,” *CoRR*, vol. abs/1811.08933, 2018. [Online]. Available: http://arxiv.org/abs/1811.08933

[136] C. Avalos Baddouh, M. Khairy, R. N. Green, M. Payer, and T. G. Rogers, “Principal Kernel Analysis: A Tractable Methodology to Simulate Scaled GPU Workloads,” in *54th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 724–737. [Online]. Available: https://doi.org/10.1145/3466752.3480100

[137] A. Gutierrez, B. M. Beckmann, A. Dutu, J. Gross, M. LeBeane, J. Kalamatianos, O. Kayiran, M. Poremba, B. Potter, S. Puthoor, M. D. Sinclair, M. Wyse, J. Yin, X. Zhang, A. Jain, and T. Rogers, “Lost in Abstraction: Pitfalls of Analyzing GPUs at the Intermediate Language

Level," in *24th IEEE International Symposium on High Performance Computer Architecture*, ser. HPCA, Feb 2018, pp. 608–619.

[138] S. Rogers, J. Slycord, M. Baharani, and H. Tabkhi, "gem5-SALAM: A System Architecture for LLVM-based Accelerator Modeling," in *53rd Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO, 2020, pp. 471–482.

[139] Z. Spencer, S. Rogers, J. Slycord, and H. Tabkhi, "Expanding Hardware Accelerator System Design Space Exploration with gem5-SALAMv2," *Journal of Systems Architecture*, vol. 154, p. 103211, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1383762124001486

[140] A. Chaudhari and M. D. Sinclair, "Toward Full-System Heterogeneous Simulation: Merging gem5-SALAM with Mainline gem5," in *6th gem5 Users' Workshop*, 6 2025.

[141] A. Smith, B. Bruce, J. Lowe-Power, and M. D. Sinclair, "Designing Generalizable Power Models For Open-Source Architecture Simulators," in *3rd Open-Source Computer Architecture Research Workshop*, ser. OSCAR, 2024.

[142] V. Ramadas, M. Poremba, B. M. Beckmann, and M. D. Sinclair, "Improving gem5's GPU FS Support," in *The 5th gem5 Users' Workshop*, 6 2023.

[143] ——, "Simulation Support for Fast and Accurate Large-Scale GPGPU and Accelerator Workloads," in *3rd Open-Source Computer Architecture Research Workshop*, ser. OSCAR, 2024.

[144] V. Ramadas and M. D. Sinclair, "Simulating Machine Learning Models at Scale," in *SRC TECHCON*, 9 2024.

[145] C. Hughes, S. D. Hammond, R. J. Hoekstra, M. Zhang, Y. Liu, and T. Rogers, "SST-GPU: A Scalable SST GPU Component for Performance Modeling and Profiling," *Sandia National Lab*, 1 2021. [Online]. Available: https://www.osti.gov/biblio/1762830

[146] C. Hughes, S. D. Hammond, M. Khairy, M. Zhang, R. Green, T. Rogers, and R. J. Hoekstra, "Balar: A SST GPU Component for Performance Modeling and Profiling," *Sandia National Lab*, 9 2019. [Online]. Available: https://www.osti.gov/biblio/1560919

[147] M. Hsieh, K. Pedretti, J. Meng, A. Coskun, M. Levenhagen, and A. Rodrigues, "SST + Gem5 = a Scalable Simulation Infrastructure for High Performance Computing," in *Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques*, ser. SIMUTOOLS '12. Brussels, BEL: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2012, p. 196–201.

[148] H. Nguyen and J. Lowe-Power, "gem5/SST Integration 2021: Scaling Full-system Simulations," in *The 4th gem5 Users' Workshop with ISCA*, 2022.

[149] M. Maiterth, W. H. Brewer, J. S. Kuruvella, A. Dey, T. Z. Islam, K. Menear, D. Duplyakin, R. Kabir, T. Patki, T. Jones *et al.*, "HPC digital twins for evaluating scheduling policies, incentive structures and their impact on power and cooling," in *SC25-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2025.

[150] W. Brewer, M. Maiterth, and D. Fay, "Trace replay simulation of MIT SuperCloud for studying optimal sustainability policies," in *2025 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE, 2025.

[151] S. Kalepu, W. H. Brewer, M. Maiterth, and R. Vuduc, "Virtual benchmarking for HPC systems using ExaDigiT and Calculon," in *2025 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE, 2025.

[152] G. von Laszewski, B. Hawks, M. Colombo, R. Shiraishi, A. Krishnan, N. Tran, and G. C. Fox, "Mlcommons science working group ai benchmarks collection," GitHub, Jun. 2025, online Collection: =https://mlcommons-science.github.io/benchmark/. [Online]. Available: https://mlcommons-science.github.io/benchmark/benchmarks.pdf