Greening AI-enabled Systems with Software Engineering: A Research Agenda for Environmentally Sustainable AI Practices

Luís Cruz
Delft University of Technology
Delft, The Netherlands
L.Cruz@tudelft.nl

João Paulo Fernandes
New York University Abu Dhabi
United Arab Emirates
jpf9731@nyu.edu

Maja H. Kirkeby*
Roskilde University
City, Country
kirkebym@acm.org

Silverio Martínez-Fernández
Universitat Politècnica de Catalunya
Barcelona, Spain
silverio.martinez@upc.edu

June Sallou†
Wageningen University & Research
The Netherlands
june.sallou@wur.nl

Hina Anwar
University of Tartu
Tartu, Estonia
hina.anwar@ut.ee

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

Justus Bogner
Vrije Universiteit Amsterdam
Amsterdam, The Netherlands
j.bogner@vu.nl

Joel Castaño
Universitat Politècnica de Catalunya
Barcelona, Spain
joel.castano@upc.edu

Fernando Castor
University of Twente
Enschede, The Netherlands
f.castor@utwente.nl

Aadil Chasmawala
New York University Abu Dhabi
United Arab Emirates
aac10066@nyu.edu

Simão Cunha
University of Minho
Braga, Portugal
a93262@alunos.uminho.pt

Daniel Feitosa
University of Groningen
Groningen, The Netherlands
d.feitosa@rug.nl

Alexandra González
Universitat Politècnica de Catalunya
Barcelona, Spain
alexandra.gonzalez.alvarez@upc.edu

Andreas Jedlitschka
Fraunhofer Institute for Experimental Software Engineering
Kaiserslautern, Germany
Andreas.Jedlitschka@iese.fraunhofer.de

Patricia Lago
Vrije Universiteit Amsterdam
Amsterdam, The Netherlands
p.lago@vu.nl

Ana Oprescu
Universiteit van Amsterdam
Amsterdam, The Netherlands
a.m.oprescu@uva.nl

Pooja Rani
University of Zurich
Zurich, Switzerland
rani@ifi.uzh.ch

João Saraiva
University of Minho & INESC TEC
Braga, Portugal
saraiva@di.uminho.pt

Federica Sarro
University College London
London, United Kingdom
f.sarro@ucl.ac.uk

Raghavendra Selvan
University of Copenhagen
Copenhagen, Denmark
raghav@di.ku.dk

Karthik Vaidhyanathan
SERC, IIIT Hyderabad
Hyderabad, India
karthik.vaidhyanathan@iiit.ac.in

Roberto Verdecchia
University of Florence
Florence, Italy
roberto.verdecchia@unifi.it

Ivan P. Yamshchikov
THWS
Würzburg, Germany
ivan.yamshchikov@thws.de

Henry Muccini
University of L'Aquila
L'Aquila, Italy
henry.muccini@univaq.it

ABSTRACT

The environmental impact of Artificial Intelligence (AI)-enabled systems is increasing rapidly, and software engineering plays a critical role in developing sustainable solutions. The "Greening AI with Software Engineering" workshop,[1] funded by the *Centre Européen de Calcul Atomique et Moléculaire* (CECAM) and the Lorentz Center, provided an interdisciplinary forum for 29 participants, from practitioners to academics, to share knowledge, ideas, practices, and current results dedicated to advancing green software and AI research. The workshop was held February 3-7, 2025, in Lausanne, Switzerland. Through keynotes, flash talks, and collaborative discussions, participants identified and prioritized key challenges for the field. These included *energy assessment and standardization*, *benchmarking practices*, *sustainability-aware architectures*, *runtime adaptation*, *empirical methodologies*, and *education*. This report presents a research agenda emerging from the workshop, outlining open research directions and practical recommendations to guide the development of environmentally sustainable AI-enabled systems rooted in software engineering principles.

*Corresponding author.

†The first five authors (alphabetic order) served as workshop organizers. The remaining authors (alphabetic order) participated in the workshop and actively contributed to the writing and review of the article.

[1]https://www.cecam.org/workshop-details/greening-ai-with-software-engineering-1358

1. INTRODUCTION

The rapid advancement of Artificial Intelligence (AI) has brought substantial benefits across numerous domains, but it has also raised growing concerns about its environmental sustainability. As AI models increase in size, complexity, and deployment scale, so do their energy consumption and carbon footprint (e.g., [27]). Training large-scale models can consume as much energy as powering homes for weeks or months, while inference at scale further compounds this impact. Yet methods for measuring, optimizing, and communicating these effects remain fragmented, inconsistent, and underdeveloped [29].

Software Engineering (SE) has a critical role to play in addressing these pressing challenges. Architectural decisions, development practices, tooling, benchmarking, and lifecycle management, all shape the environmental profile of AI-enabled systems. However, sustainable AI development is inherently interdisciplinary, requiring coordination across AI research, systems engineering, education, and policy. To understand how SE can contribute meaningfully to this space, we must examine methodological practices, educational strategies, tools, and infrastructure-level design. As such this article is a response to the call for contributions to address emerging trends from an SE perspective [12].

This article presents key insights from the "Greening AI with Software Engineering" workshop held in Lausanne, Switzerland, on February 3-7, 2025, co-financed by the Centre Européen de Calcul Atomique et Moléculaire (CECAM) and the Lorentz Center. The workshop gathered 29 participants from academia and industry for an intensive program of keynotes, flash talks, and breakout discussions. Designed to support co-creation, the workshop format enabled participants to collaboratively identify and refine the focus areas that structure this report.[2] In doing so, the workshop surfaced open questions, practical tensions, and recurring patterns in how AI sustainability is approached across domains.

The five focus areas presented in this report are: 1) energy assessment and standardization, 2) evaluation and benchmarking of AI sustainability, 3) software architecture and lifecycle, 4) empirical methods and reproducibility, and 5) education and awareness. These areas structure the main body of the article and reflect both the breadth and depth of sustainability challenges discussed during the workshop. Across the five focus areas, we identify cross-cutting concerns such as *standardization*, *metric design*, and *holistic thinking* that shape the broader research agenda. By summarizing and synthesizing these discussions, this article contributes to building a shared vocabulary of key terms, distinctions, and recurring themes, and outlines a research agenda for advancing environmentally sustainable AI with Software Engineering. A particular effort has been made to emphasize how we imagine Software Engineering can contribute across these five areas.

2. ENERGY ASSESSMENT AND STANDARDIZATION

Accurately assessing the energy consumption of AI-enabled systems is a foundational step toward understanding and improving their environmental impact. However, the current landscape of energy measurement and estimation is fragmented, with a lack of or inconsistent tools, granularity levels, and reporting practices. This section outlines the main challenges associated with measuring energy usage in AI-enabled systems.

2.1 Bridging Granularity and Standardization in Energy Metrics

Achieving high-level standards for collecting system-wide energy footprint data requires a standardized approach similar to the Intel's Running Average Power Limit (RAPL)[3], but one that applies uniformly across all computational devices. Standardization efforts prioritize broad applicability across entire software systems rather than precision at the component level. However, organizations require a more granular perspective on energy consumption to support informed decision-making.

Establishing high-level standards would enable the collection of consistent and comparable metrics across diverse computing environments. A holistic methodology should capture energy consumption data at multiple granularities, from system-wide reporting to fine-grained scientific measurements, ensuring traceability across levels. A standardized approach should facilitate comparing different solutions, enabling consistent evaluation of their resource demands across computing devices. It should support precise measurements when needed, such as analyzing short-lived executions.

These objectives align with ongoing research on energy consumption in AI and Machine Learning (ML), as seen in studies such as Green AI [40], Carbontracker [3], and research on estimating the carbon footprint of large-scale AI models [25].

2.2 Methodological Consistency

Assessing energy consumption of an ML model is inherently challenging because a model's behavior depends on multiple factors, including architecture, platform, hyperparameters, quantization level, task type, and input/output characteristics. Additionally, ML models exhibit non-deterministic execution, making reproducibility a significant challenge. Since testing every possible combination of influencing variables is infeasible, identifying key sources of variation is essential to obtain meaningful and transferable results.

Unlike program execution time, the energy consumption of ML models is difficult to explain. Without robust methodologies, assessments risk being inaccurate, inconsistent, or non-generalizable across different contexts. Ensuring accuracy, reliability, and reproducibility is crucial to making energy measurements reflect real-world behavior as closely as possible.

Addressing this challenge requires balancing abstraction and precision – developing a methodology that is both widely applicable and detailed enough to capture ML models' energy behavior accurately. This involves distinguishing between factors that significantly impact energy consumption and those that can be abstracted away without compromising assessment validity. For instance, on the one hand, studies suggest that in code generation

[2]These focus areas emerged through an iterative process: The participants first contributed individual reflections using sticky notes, which were grouped into preliminary clusters. These were refined through rotating group discussions supported by anchors who maintained continuity and captured insights. Then, the participants committed to a single area and co-developed draft texts. After the workshop, the coordinators consolidated the material, removed underdeveloped topics, and identified cross-cutting themes. A draft report was shared with the participants and revised based on their feedback.

[3]`https://www.intel.com/content/www/us/en/developer/articles/technical/software-security-guidance/advisory-guidance/running-average-power-limit-energy-reporting.html`

with large language models, output length influences energy consumption more than input length [1]. On the other hand, in tasks such as code completion, the large amount of context required to generate small pieces of code may result in a shift in this balance. Another promising approach is using mini-models as proxies for larger models, leveraging evidence that smaller models with the same architecture can effectively estimate the energy footprint of their larger counterparts [34].

Due to the heterogeneity and continuous evolution of AI-enabled systems, a single, rigid methodology for energy assessment is inadequate. The diversity of models, architectures, hardware, and workloads prevents any universal approach from accurately capturing energy consumption in all cases. Moreover, as AI technologies evolve, current methodologies may become obsolete. This presents two key challenges: first, providing practitioners with reliable guidance in selecting appropriate energy assessment methods, and second, ensuring that methodologies remain adaptable as AI-enabled systems continue to change.

To address these challenges, a flexible yet structured approach is essential. Instead of enforcing a single assessment framework, practitioners need adaptable guidelines that enable them to navigate the trade-offs between different methodologies while ensuring compliance with high-level energy assessment standards. One potential solution is the creation of a *publicly accessible repository* of energy assessment methodologies, detailing their assumptions, scope, and typical use cases, similarly to a family of software patterns [14]. This would assist practitioners in understanding the available options and making informed decisions based on their system's characteristics and objectives. Additionally, establishing clear selection criteria could provide a structured way to evaluate methodologies, helping practitioners assess trade-offs between factors such as hardware efficiency, hyperparameter tuning, and overall system-wide energy consumption.

Another important consideration is the *interoperability between assessment methodologies*. Many existing approaches focus on isolated measurements, but there may be value in designing methodologies that can interface with one another, creating a more modular and adaptable assessment pipeline. This would facilitate comparisons across models, architectures, and computing environments, while also promoting a degree of standardization. Furthermore, as AI-enabled systems continue to evolve, mechanisms for methodology migration may become necessary to ensure assessments remain relevant and comparable over time. Developing protocols for transitioning between methodologies could help maintain continuity in energy assessments, even as best practices evolve.

At a broader level, the concept of a meta-methodology –a framework for guiding decision-making in energy assessment– could be worth exploring. Previous work [17, 32] in the area of Software Engineering has leveraged this approach to help identify energy consumption hotspots in software systems. Rather than prescribing a single solution, a meta-methodology could help practitioners structure their approach based on the priorities relevant to their specific use case. For instance, it could offer guidance on whether capturing the energy cost of hyperparameter tuning is more relevant than measuring precise hardware consumption, depending on the context. These ideas build on ongoing research into ML energy consumption. Existing studies have explored the energy costs of deep learning frameworks [2, 20, 15], the impact of batching strategies on inference efficiency [47], and the role of quantization techniques in energy efficiency [36]. By integrating insights from this work into a more flexible and evolving framework, it may be possible to develop energy assessment methodologies that remain rigorous and adaptable as AI technologies continue to evolve.

2.3 Positioning Software Engineering

Software engineering advances energy assessment in AI by enabling system-level and multi-granular measurement approaches that move beyond isolated model evaluation. We imagine that it will provide generalised structured yet flexible methodologies – meta-methodologies and adaptable guidelines–that support context-sensitive assessments across heterogeneous platforms. Additionally, Software Engineering approaches may contribute to the standardization and interoperability of tools and metrics, making energy measurements reproducible and comparable across systems and over time.

3. EVALUATING AI FOR SUSTAINABILITY

Assessing the energy consumption of AI-enabled systems requires *benchmarking frameworks* that incorporate both energy and carbon footprint metrics, alongside conventional measures such as accuracy, latency, and throughput. The objective is to develop methodologies that are widely adoptable and easily integrated into real-world CI/AI pipelines while being either hardware-aware or hardware-agnostic through normalization. These frameworks should account for both training and inference energy consumption, ensuring that energy assessments capture the full lifecycle of an AI model.

The need for such benchmarks originates from several pressing concerns. The environmental impact of AI is growing, with large-scale models consuming substantial energy and raising sustainability challenges ([48, 5]). All decision makers, such as developers, organizations, and policymakers, require reliable, meaningful, and comparable energy metrics (e.g., Joules, CO_2eq.) to make informed trade-offs between functional and extra-functional properties, such as accuracy, security, runtime, and energy cost. Additionally, potential regulatory requirements (e.g., the EU AI Act, ESG, and CSRD) may mandate energy reporting for AI-enabled systems. From an economic perspective, energy-efficient AI models offer business advantages, particularly in reducing operational costs in data centres and edge deployments. As AI regulation and standardization efforts evolve, energy measurement could become a critical service for evaluating AI models.

3.1 Standardized Benchmarking for AI Sustainability

A key step toward sustainable AI is the extension of existing benchmarking frameworks to include standardized energy and carbon metrics. Well-established benchmarking suites such as MLPerf [38, 28] or Hugging Face[4] leaderboards provide a structured foundation for measuring model performance, but currently with very limited standardized energy reporting. Extending these benchmarks to incorporate energy consumption across different layers of AI models, from individual operations to end-to-end pipelines, would enable a more comprehensive assessment framework.

Beyond energy measurement, integration with interdisciplinary research and policy standards is critical for ensuring broad adoption. Efforts such as the impact framework of the Green Software Foundation (.imp files) provide potential pathways for aligning AI benchmarking with emerging sustainability regulations. Additionally, defining a common ontology for AI energy benchmarking

[4] `https://huggingface.co/`

would facilitate interoperability across research domains and enable collaboration between computer scientists, policymakers, and industry stakeholders.

At the algorithmic level, energy-efficient neural architecture search (NAS) methods such as EC-NAS [5] and CE-NAS [48] have demonstrated the potential for optimizing architectures based on energy and carbon considerations. Similarly, approaches like Once-for-All (OFA) [9], which train a single network that can be specialized for different latency and energy constraints, illustrate the feasibility of integrating energy-awareness into AI model development.

3.2 Multi-Dimensional Evaluation

To ensure a holistic assessment of AI sustainability, energy and carbon footprint metrics must be evaluated alongside traditional performance measures such as accuracy and latency. A multi-dimensional evaluation framework would provide a clearer picture of trade-offs between computational cost and model effectiveness.

A key aspect of this evaluation is the amortization of training energy costs over inference usage. Training large AI models is energy-intensive, but its impact on sustainability depends on how frequently the model is used. By treating training energy as a capital expense and distributing it across the total number of expected inferences, researchers can compute an amortized energy cost per inference, offering a more balanced sustainability metric.

Several existing frameworks already explore partial energy reporting. Hugging Face Model Cards sometimes include throughput or energy-related metrics, but these are not consistently standardized. Tools such as IrEne [10] and Smaragdine [4] provide entry points for AI energy accounting, but there is a need for a more structured methodology that aligns with AI benchmarking standards.

3.3 Hardware and Normalized Metrics

Ensuring that energy benchmarks are hardware-agnostic is essential for fair and meaningful comparisons across different AI models and deployment environments [8]. However, hardware variations significantly impact energy consumption, making it challenging to compare models trained and deployed on different infrastructure.

One potential approach is to measure floating point operations (FLOPs) and integrate them into energy-prediction models, allowing researchers to estimate energy usage independently of specific hardware configurations. However, it is known that all FLOPs are not equal [24], and directly relying on FLOPs-driven energy measurements could bias these metrics [19]. Additionally, defining a baseline hardware profile could facilitate normalized energy reporting, ensuring that comparisons between models remain consistent regardless of hardware disparities.

Beyond FLOPs, reporting on broader resource utilization metrics (e.g., memory access patterns, I/O usage, and GPU/TPU power draw) could provide a more complete picture of energy consumption [3]. A standardized methodology for energy-aware reporting across AI hardware platforms would help ensure that benchmarks remain interpretable and broadly applicable.

Interdisciplinary collaborations between computer scientists, hardware engineers, and regulatory bodies could further strengthen AI benchmarking efforts by developing guidelines that integrate environmental sustainability concerns directly into AI deployment practices [45].

3.4 Positioning Software Engineering

We imagine software engineering to advance Green AI by structuring how sustainability is evaluated and reported. Establishing and adopting standardized energy efficiency benchmarks may guide developers in evaluating, comparing, and optimizing the environmental impact of AI-enabled systems. This includes integrating energy and carbon metrics into evaluation pipelines alongside accuracy and latency, applying normalization techniques to ensure fair cross-platform comparisons, and using lifecycle-aware models—such as amortized training cost—to contextualize energy consumption over time and usage scale.

4. SYSTEM ARCHITECTURE AND LIFECYCLE

For AI-enabled systems, sustainability-driven architecting should integrate sustainability principles related to both lifecycle management (process) and architecture design (product). This means that sustainability-driven design decisions and quality assessment results should be considered throughout all phases of the lifecycle of AI-enabled software systems. Typical software architecture concepts [6] – such as architectural tactics, scenarios, patterns, practices, indicators, metrics, and measures – should be traceable to the corresponding AI-enabled architectural elements. Additionally, sustainability learning should provide guidance for dynamic adaptation in both MLOps and AI-enabled architectures.

4.1 Architecting for Sustainable AI

Current research on environmentally sustainable AI tends to emphasize improving the energy efficiency of individual models. However, AI-enabled systems often comprise multiple components – both AI and non-AI ones – that interact in complex ways. A narrow focus on isolated model performance may obscure the broader architecture context that ultimately determines system-wide sustainability [45]. Adopting a system-level perspective enables sustainability considerations to be embedded throughout the AI-enabled architecture, from early design and deployment to runtime operation and long-term evolution.

From an architectural perspective, AI models function as specialized components that expose interfaces, interact with surrounding services, and influence the system's overall quality attributes [23]. Established software architecture practices – such as modular design, scenario-based evaluation, and trade-off analysis – can guide the integration of AI components in ways that align with broader sustainability goals. Rather than selecting the most powerful model by default, architects should be able to consider which model best satisfies the system's architecturally significant requirements [13], balancing accuracy, latency, and energy consumption.

Several design strategies can support these goals [21]. For example, selecting models that meet only the necessary performance requirements avoids energy waste due to overprovisioning [43]. Implementing lightweight models on the client side, with fallback to more complex models on the server, can improve efficiency in distributed deployments [46]. Connector logic that dynamically routes requests based on resource constraints can further enhance adaptability [42, 31]. These decisions can be informed by predictive tools that estimate energy consumption at design time, helping architects evaluate tradeoffs early in the process.

In distributed AI-enabled systems, executing model workloads across edge, fog, and cloud layers enables more granular control of resource allocation and energy use. Refactoring legacy architectures using energy-aware tactics, and documenting architectural decisions alongside their energy implications, can strengthen

traceability and operational transparency. Together, these practices enable the development of AI-enabled systems that prioritize sustainability without compromising performance, reliability, or maintainability across their lifecycle.

4.2 Green AI at Runtime

While training large AI models is energy-intensive, inference can account for a significant share of the system's total energy consumption, particularly when deployed at scale [27]. Addressing sustainability at runtime is therefore critical, yet remains less explored than energy-aware training practices. AI-enabled systems operate under diverse and often unpredictable runtime conditions, including changes in data quality, model performance, business needs, and infrastructure availability. These uncertainties impact both energy efficiency and quality of service.

To manage this complexity, runtime strategies must balance adaptability with sustainability. Enhancing observability – the ability to monitor system behavior and internal states – supports energy-aware decision-making by making deviations from expected performance and consumption patterns visible. In uncertain environments, observability enables early detection of inefficiencies and supports dynamic reconfiguration.

Self-adaptation mechanisms offer another path to improving runtime sustainability. By adjusting system structure or behavior in response to observed conditions, self-adaptive systems can optimize trade-offs between energy use and other quality attributes. Examples include dynamically selecting between models of varying complexity, reallocating workloads between edge and cloud resources, or retraining models on lower-energy hardware. These strategies are particularly valuable in MLOps pipelines, where frequent model updates and changing deployment contexts can otherwise lead to unnecessary energy costs.

Tools such as sustainability decision maps [22] can further support runtime reasoning by providing structured representations of trade-offs and system goals. These maps help track alignment between energy-related targets and system behavior over time, enabling more informed and transparent decisions. Integrating such tools into MLOps workflows can improve feedback loops and support continuous energy optimization [7]. Runtime sustainability requires more than localized model improvements –it calls for infrastructure-aware, adaptive, and transparent system-level coordination. Addressing energy use at this stage seems essential to ensuring that the long-term operation of AI-enabled systems remains aligned with sustainability goals.

4.3 Systemic Sustainability in MLOps Workflows

Sustainability efforts in AI development are often hindered by limited collaboration and information sharing across roles in the MLOps lifecycle. Although various metrics and logs are collected throughout model training, deployment, and monitoring, these data are rarely used holistically to inform sustainability-oriented decisions. Decisions made in isolation – such as optimizing training energy while neglecting inference costs – can lead to counterproductive outcomes when viewed from a system-wide perspective.

Improved traceability of energy-relevant decisions is essential to enable meaningful collaboration. Sustainability-related architectural elements – such as design patterns, quality attributes, and runtime adaptations – should be linked to observable data across the MLOps stack. Mechanisms for communication and alignment – such as APIs that surface sustainability metrics between pipeline components – can support consistent decision-making and make trade-offs across lifecycle phases more transparent.

A central challenge, however, is the lack of clearly defined responsibilities for sustainability. It is often unclear whether accountability lies with infrastructure providers, system architects, AI developers, or software engineers. This ambiguity prevents coordinated efforts to assess and improve sustainability across the full pipeline. Clarifying roles and responsibilities is a prerequisite for embedding sustainability into each stage of the development process.

Distinguishing between Green AI and Green SE may further support collaboration. While Green AI focuses on improving the energy efficiency of AI models themselves, Green SE emphasizes the sustainability of the full system in which those models are embedded. Clarifying this distinction can help assign accountability more effectively and ensure that sustainability is treated as a system-wide concern rather than the isolated optimization of a single model.

4.4 Positioning Software Engineering

We imagine software engineering to advance Green AI by treating AI as a software system component. By considering environmental sustainability across the entire life cycle including design and runtime, we adopt a holistic view on the sustainability impact of AI.

Software engineering approaches include modular design, component wise traceability, and runtime adaptability, which enable dynamic optimization of energy use across deployment contexts and division of responsibility.

5. EMPIRICAL EVALUATION

Research activity in Green AI is expanding rapidly, requiring a growing commitment to evidence-based approaches for enhancing the sustainability of AI-enabled systems. Empirical studies are increasingly used to investigate the environmental impacts of AI development and deployment [15, 47, 25, 27, 37, 35, 16, 26] and to evaluate the level of improvement regarding sustainability of developed solutions achieved through newly proposed methods.

This growing body of empirical work has the potential to contribute to a shared knowledge base, enabling stakeholders to move beyond anecdotal evidence and toward informed, data-driven decision-making. Aligning methods and reporting practices allows better comparability between studies and facilitates the identification of robust patterns and best practices.

5.1 Actionable Body of Knowledge

Although empirical methods are becoming central to Green AI research, the results remain difficult to compare or generalize. This is largely due to methodological inconsistencies, varied study designs, and context-specific reporting. While methods such as controlled experiments and case studies are available, they are applied unevenly, limiting the ability to synthesize findings across studies. This has led to a body of evidence that is difficult to accumulate or systematically synthesize.

A long-term vision for the field is to establish a coherent and evolving body of knowledge that allows stakeholders to make evidence-based decisions about sustainability practices in AI development and deployment. Such a shared foundation would enhance the comparability and generalizability of results and support the creation of benchmarks, inform regulatory efforts, and accelerate the

transfer of knowledge from research to practice.

Realizing this vision requires coordinated efforts across the research community to align with empirical standards, promote replication, and foster shared methodological frameworks. Without such alignment, the field risks fragmentation and limited impact. As Hart and Baehr [18] argue, a sustainable body of knowledge in technical domains relies on both taxonomic structures and community engagement. The empirical Green AI community must thus evolve a structured landscape of open questions, replicated evidence, and reliable methodologies that can guide both researchers and practitioners.

5.2 Standardized and Reliable Metrics

Sustainability assessments in software systems often rely on metrics that are inconsistently defined and unevenly applied, even when targeting comparable phenomena such as energy use or execution time. These inconsistencies emerge from a range of factors, including variation in hardware configurations, abstraction levels, and data collection tools [47, 15]. Consequently, findings from different studies often resist meaningful comparison.

Such inconsistencies highlight a broader challenge: the lack of standardized and reliable metrics, which continues to obstruct efforts toward cumulative progress in Green AI research [3, 25]. Metrics used in this field span various abstraction levels, from low-level physical measures such as energy per instruction, to high-level proxies such as runtime or task throughput, and to more integrative compositions like sustainability scores [41, 11] that, for instance, incorporate accuracy and explainability [36]. However, without consensus on their definitions, reliability, and the appropriate contexts for use, these metrics yield incompatible results that hinder replication and meta-analysis. Addressing this challenge would not only allow for systematic comparison between studies but would also provide a foundation for evidence-based guidelines and policy recommendations. It would support the identification of effective green practices and enable transparent and consistent reporting across academic and industrial settings [27].

One possible direction is to develop a taxonomy of metrics that distinguishes between direct, proxy, and composite types. This taxonomy would ideally be supported by an articulation of each metric's assumptions, limitations, and the contexts in which they are most appropriate. Among composite metrics, a further distinction may be drawn between multi-dimensional and compositional approaches: while the former preserve trade-offs by reporting multiple values along separate axes (e.g., energy, latency, accuracy), the latter collapse these into a single score to support optimization or policy-facing decision-making. As understanding in the field evolves, so too might the criteria for empirical validation. There may also be value in exploring how standardized protocols for measurement can be developed and adapted over time – particularly in methods such as case studies or surveys, where such practices are not yet established. Crucially, the standardization of metrics must be balanced with openness to innovation. New metrics may emerge in response to evolving technologies or new sustainability concerns. To manage this balance, a community-driven oversight mechanism – perhaps in the form of a metrics committee – could evaluate, endorse, and curate both existing and emerging measurement practices.

Lessons can be taken from the empirical software engineering community [44]. Further insights might be drawn from search-based software engineering research, where trade-off measurements and analysis have been well studied [39], as well as from health research, where standardized data collection and reporting protocols have long provided a foundation for reproducibility and comparability [18].

5.3 Positioning Software Engineering

We imagine software engineering to enhance Green AI by contributing empirical evaluation practices that enable trustworthy and generalizable sustainability claims. Drawing from empirical software engineering, we may adopt principles of methodological rigor, reproducibility, and structured comparison to support the development of a coherent evidence base. Software Engineering may also guide the creation of standardized metric taxonomies –distinguishing between direct, proxy, and composite measures– and alignment of data protocols across tools and hardware. We envision that empirical research within Green AI and Green Software Engineering will continue to cross-fertilize, strengthening both fields.

6. EDUCATION AND AWARENESS

Education and awareness play a crucial role in fostering sustainable AI practices. However, several challenges need to be addressed in order to effectively integrate Green AI principles into curricula and industry training programs. Below, we highlight key challenges and considerations grouped by stage in the educational process.

Embedding sustainability into SE and AI education requires coordinated attention across several layers of the educational process [33, 30]. At the curriculum level, integrating sustainability meaningfully requires institutional commitment and long-term thinking. While sustainability is inherently a long-term concern, academic structures often prioritize short-term outcomes. Institutions can leverage Intended Learning Outcomes (ILOs) to systematically integrate sustainability into educational goals, but must also be mindful of avoiding superficial virtue signalling. Gradual introduction of sustainability topics into existing, mandatory courses can increase reach beyond the self-selecting students typically drawn to electives.

Designing and implementing effective courses on sustainability-aware SE and AI poses practical barriers. Many Green AI courses rely on a wide array of prerequisites, spanning hardware, software, and systems-level topics. Hands-on activities are constrained by the need for specialized tools and equipment, and the lack of mature, accessible benchmarking frameworks. Furthermore, connecting energy consumption with more intuitive performance measures like time can be non-trivial. The design of assignments must also reflect the complexity of moving from energy assessment to actionable improvements, given the many interacting variables at play. To support effective learning, example systems used in teaching should be neither trivial nor overwhelming–ideally open-source and industry-relevant.

Students must navigate a high cognitive load when learning to develop sustainable software. The architectural concepts involved– such as patterns, tactics, styles, quality attributes, and trade-offs– require deep understanding and differentiation. Thinking in terms of lifecycle sustainability adds further complexity, requiring students to reason across temporal and system boundaries. In addition, motivating students to value sustainability as a critical, multidimensional quality attribute is a challenge, particularly in industrial contexts where it is often not prioritized.

To assess the impact of sustainability teaching, educators must

draw on robust pedagogical methods. Concepts like the Jevons Paradox, which highlights the counterintuitive effects of improved efficiency on total consumption, can be used to encourage critical thinking. Empirical evaluation approaches, such as A/B testing of teaching interventions, can help demonstrate effectiveness and inform iteration. Tying these evaluation methods to the intended learning outcomes supports the alignment between teaching goals and assessment practice.

These educational efforts are further complicated by the immaturity of the research field itself. In domains like Green AI, much of the work is mainly experimental and rapidly evolving. This makes it difficult to develop stable and reusable teaching packages. Moreover, translating low-level energy measurements into metrics that are meaningful for both technical and non-technical audiences remains an open challenge, requiring ongoing engagement from both educators and researchers.

6.1 Aligning Software Engineering and AI

We imagine that software engineering and AI education can both benefit from each other in our efforts to promote sustainability. While sustainability is not yet a core element in either field, SE's system-level thinking and AI's data-centric methods offer complementary strengths. By learning from each discipline's approaches, we may grow educational practices that embed environmental awareness more deeply. We envision this exchange as a path toward a shared sustainability culture that strengthens both disciplines.

7. CROSS-CUTTING REFLECTIONS

Building on the synthesis provided in the discussion, this section identifies recurring themes that emerged across the workshop's diverse focus areas. These themes–*standardization*, *metrics*, and *holistic and longitudinal thinking*– cut across the topical boundaries of energy assessment, sustainability evaluation, architecture, empirical methodology, and education. They represent significant concerns for the development of environmentally sustainable AI.

Standardization was repeatedly highlighted as a fundamental need in measurement practices, benchmarking efforts, and empirical reporting. Discrepancies in the way energy or efficiency is defined (for example, joules, runtime, or CO_2) currently hinder comparability and prevent coordination between research, industry, and policy stakeholders. In the context of benchmarking frameworks, the lack of standardized reporting of energy and carbon consumption remains a major limitation (Sect. 3.2). Similarly, proposals for normalized reporting across heterogeneous hardware platforms point to the necessity of standardization for interpretability and fairness (Sect. 3.3). The development of a taxonomy of metrics, protocols for their application, and oversight mechanisms is central to advancing shared practices (Sect. 5.2).

Metrics themselves constitute both a technical and conceptual challenge. As highlighted in the discussion on hardware-normalized metrics (Sect. 3.3) and architectural decision-making (Sect. 4), sustainability cannot be captured by a single scalar value. Multi-dimensional trade-offs, between accuracy, energy, latency, and explainability, require carefully chosen metrics. These must be context-sensitive yet comparable across cases. Empirical work has begun to surface classifications of direct, proxy, and composite metrics, but their assumptions and limitations remain under articulated (Sect. 5.2).Developing metrics that are both robust and adaptable is essential for meaningful evaluation and progress.

Holistic and longitudinal thinking emerged as a critical counterbalance to narrow or isolated optimization efforts. Lifecycle sustainability –spanning design, training, deployment, and operation– was emphasized in architecture discussions (Sect. 4) and sustainability evaluation frameworks (Sect. 3.2). However, workshop contributions also pointed to the importance of longer-term effects: how current benchmarking norms or energy optimizations may influence future systems and behaviors. The underutilization of longitudinal studies in empirical work and the potential for counterproductive outcomes such as the Jevons paradox (Sect. 6) underscore the need to assess not only a system's immediate footprint but its place within a larger temporal trajectory [45]. Incorporating this perspective is particularly relevant for decision-making in MLOps contexts, where short-term gains may obscure longer-term sustainability costs (Sect. 4.3).

Together, these themes suggest that advancing sustainable AI through software engineering requires more than isolated technical solutions. It demands a coordinated effort to establish shared foundations, articulate meaningful metrics, and reason across both system and temporal boundaries.

8. CONCLUSION

This article presented a synthesis of discussions from the "Greening AI with Software Engineering" workshop, which brought together researchers and practitioners to explore the environmental sustainability of AI through a software-centric lens. Rather than prescribing a single approach or framework, the contributions reflect a shared recognition that addressing AI's environmental impact requires attention to multiple, interconnected dimensions.

The report is structured around five key areas: energy assessment and standardization, evaluation and benchmarking, architecture and lifecycle design, empirical methods and reproducibility, and education and awareness. Although each area raises distinct challenges, the discussions also revealed three cross-cutting themes–standardization, metrics, and holistic system thinking–that connect these areas and shape a broader research agenda.

What emerged is not a finished blueprint, but a conceptual research agenda: a set of aligned concerns, unresolved tensions, and promising directions. Moving forward, the research community is invited to build on these directions by developing shared benchmarks, improving observability and traceability, refining architectural practices, and embedding sustainability into both empirical studies and software engineering education. Addressing these challenges will require not only technical innovation, but also collaboration, openness, and long-term commitment from academia and industry. By articulating these challenges and points of convergence, this article contributes to an evolving foundation for the development of environmentally sustainable AI with software engineering.

9. ACKNOWLEDGEMENT

We would like to thank the *Centre Européen de Calcul Atomique et Moléculaire* (CECAM) and the Lorentz Center for their supporting the "Greening AI with Software Engineering" workshop, which served as the foundation for the collaborative research summarized in this report.

10. REFERENCES

[1] Negar Alizadeh, Boris Belchev, Nishant Saurabh, Patricia Kelbert, and Fernando Castor. Language models in software development tasks: An experimental analysis of energy and accuracy. 2025. To appear in the Proceedings of the 22nd

International Conference on Mining Software Repositories, MSR 2025, Ottawa, Canada, April 28-29, 2025.

[2] Negar Alizadeh and Fernando Castor. Green ai: a preliminary empirical study on energy consumption in dl models across different runtime infrastructures. In *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI*, CAIN '24, page 134–139, New York, NY, USA, 2024. ACM.

[3] Lasse F. Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. 2020. Presented at ICML Workshop on Challenges in Deploying and Monitoring Machine Learning Systems.

[4] Timur Babakol and Yu David Liu. Tensor-aware energy accounting. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, ICSE '24, New York, NY, USA, 2024. ACM.

[5] Pedram Bakhtiarifard, Christian Igel, and Raghavendra Selvan. Ec-nas: Energy consumption aware tabular benchmarks for neural architecture search. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2024, Seoul, Republic of Korea, April 14-19, 2024*, pages 5660–5664. IEEE, 2024.

[6] Len Bass, Paul Clements, and Rick Kazman. *Software Architecture in Practice*. Addison-Wesley Professional, Westford, MA, USA, 3rd edition, 2012. Publication Title: The SEI Series in Software Engineering.

[7] Hiya Bhatt, Shrikara Arun, Adyansh Kakran, and Karthik Vaidhyanathan. Towards Architecting Sustainable MLOps: A Self-Adaptation Approach . In *2024 IEEE 21st International Conference on Software Architecture Companion (ICSA-C)*, pages 179–182, Los Alamitos, CA, USA, June 2024. IEEE.

[8] Lucía Bouza, Aurélie Bugeau, and Loïc Lannelongue. How to estimate carbon footprint when training deep learning models? a guide and review. *Environmental Research Communications*, 5(11):115014, nov 2023.

[9] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. 2020.

[10] Qingqing Cao, Yash Kumar Lal, Harsh Trivedi, Aruna Balasubramanian, and Niranjan Balasubramanian. Irene: Interpretable energy prediction for transformers. 2021.

[11] Iffat Fatima, Patricia Lago, Vasilios Andrikopoulos, and Bram van der Waaij. Using sustainability impact scores for software architecture evaluation. 2025. To appear in the Proceedings of the 22nd International Conference on Software Architecture (ICSA).

[12] Stefano Forti, Uwe Breitenbücher, and Jacopo Soldani. Trending topics in software engineering. *SIGSOFT Softw. Eng. Notes*, 47(3):20–21, July 2022.

[13] Xavier Franch, Silverio Martínez-Fernández, Claudia P. Ayala, and Cristina Gómez. Architectural Decisions in AI-Based Systems: An Ontological View. In Antonio Vallecillo, Joost Visser, and Ricardo Pérez-Castillo, editors, *Quality of Information and Communications Technology*, volume 1621, pages 18–27. Springer International Publishing, Cham, 2022. Series Title: Communications in Computer and Information Science.

[14] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc., USA, 1995.

[15] Stefanos Georgiou, Maria Kechagia, Tushar Sharma, Federica Sarro, and Ying Zou. Green ai: do deep learning frameworks have different costs? In *Proceedings of the 44th International Conference on Software Engineering*, ICSE '22, page 1082–1094, New York, NY, USA, 2022. ACM.

[16] Jingzhi Gong, Sisi Li, Giordano d'Aloisio, Zishuo Ding, Yulong Ye, William B. Langdon, and Federica Sarro. Greenstableyolo: Optimizing inference time and image quality of text-to-image generation. In Gunel Jahangirova and Foutse Khomh, editors, *Search-Based Software Engineering*, pages 70–76, Cham, 2024. Springer Nature Switzerland.

[17] Shuai Hao, Ding Li, William G. J. Halfond, and Ramesh Govindan. Estimating mobile application energy consumption using program analysis . In *2013 35th International Conference on Software Engineering (ICSE)*, pages 92–101, Los Alamitos, CA, USA, May 2013. IEEE.

[18] Hillary Hart and Craig Baehr. Sustainable practices for developing a body of knowledge. *Technical Communication*, 60(4):259–266, November 2013.

[19] Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research*, 21(248):1–43, 2020.

[20] Vitor Maciel Fontes Jacques, Negar Alizadeh, and Fernando Castor. A study on the battery usage of deep learning frameworks on ios devices. In *Proceedings of the IEEE/ACM 11th International Conference on Mobile Software Engineering and Systems*, MOBILESoft '24, page 1–11, New York, NY, USA, 2024. ACM.

[21] Heli Järvenpää, Patricia Lago, Justus Bogner, Grace Lewis, Henry Muccini, and Ipek Ozkaya. A Synthesis of Green Architectural Tactics for ML-Enabled Systems. In *Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Society*, pages 130–141, Lisbon Portugal, April 2024. ACM.

[22] Patricia Lago. Architecture Design Decision Maps for Software Sustainability . In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*, pages 61–64, Los Alamitos, CA, USA, May 2019. IEEE.

[23] Grace A. Lewis, Ipek Ozkaya, and Xiwei Xu. Software Architecture Challenges for ML Systems . In *2021 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 634–638, Los Alamitos, CA, USA, October 2021. IEEE.

[24] Francisco López, Lars Karlsson, and Paolo Bientinesi. Flops as a discriminant for dense linear algebra algorithms. In *Proceedings of the 51st International Conference on Parallel Processing*, ICPP '22, New York, NY, USA, 2023. ACM.

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

[26] Sasha Luccioni, Boris Gamazaychikov, Emma Strubell, Sara Hooker, Yacine Jernite, Carole-Jean Wu, and Margaret Mitchell. Ai energy score leaderboard - february 2025, feb 2025.

[27] Sasha Luccioni, Yacine Jernite, and Emma Strubell. Power Hungry Processing: Watts Driving the Cost of AI Deployment? In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '24, page 85–99, New York, NY, USA, 2024. ACM.

[28] Peter Mattson, Christine Cheng, Gregory Diamos, Cody

Coleman, Paulius Micikevicius, David Patterson, Hanlin Tang, Gu-Yeon Wei, Peter Bailis, Victor Bittorf, et al. MLPerf Training Benchmark. *Proceedings of Machine Learning and Systems*, 2:336–349, 2020.

[29] Tim Menzies and Brittany Johnson. Powering Down: An Interview With Federica Sarro on Tackling Energy Consumption in AI-Powered Software Systems . *IEEE Software*, 41(05):89–92, September 2024.

[30] Ana Moreira, Patricia Lago, Rogardt Heldal, Stefanie Betz, Ian Brooks, Rafael Capilla, Vlad Constantin Coroamă, Leticia Duboc, João Paulo Fernandes, Ola Leifler, Ngoc-Thanh Nguyen, Shola Oyedeji, Birgit Penzenstadler, Anne-Kathrin Peters, Jari Porras, and Colin C. Venters. A roadmap for integrating sustainability into software engineering education. *ACM Trans. Softw. Eng. Methodol.*, 34(5), May 2025.

[31] Nienke Nijkamp, June Sallou, Niels Van Der Heijden, and Luís Cruz. Green AI in Action: Strategic Model Selection for Ensembles in Production. In *Proceedings of the 1st ACM International Conference on AI-Powered Software*, pages 50–58, Porto de Galinhas Brazil, July 2024. ACM.

[32] Wellington Oliveira, Renato Oliveira, Fernando Castor, Gustavo Pinto, and João Paulo Fernandes. Improving energy-efficiency by recommending Java collections. *Empir. Softw. Eng.*, 26(3):55, 2021.

[33] Anne-Kathrin Peters, Rafael Capilla, Vlad Constantin Coroamă, Rogardt Heldal, Patricia Lago, Ola Leifler, Ana Moreira, João Paulo Fernandes, Birgit Penzenstadler, Jari Porras, and Colin C Venters. Sustainability in Computing Education: A Systematic Literature Review. *ACM Trans. Comput. Educ.*, 24:1–53, 2024.

[34] Haoran Qiu, Weichao Mao, Archit Patke, Shengkun Cui, Saurabh Jha, Chen Wang, Hubertus Franke, Zbigniew T. Kalbarczyk, Tamer Başar, and Ravishankar K. Iyer. Power-aware deep learning model serving with μ-serve. In *Proceedings of the 2024 USENIX Conference on Usenix Annual Technical Conference*, USENIX ATC'24, USA, 2024. USENIX Association.

[35] Saurabhsingh Rajput, Maria Kechagia, Federica Sarro, and Tushar Sharma. Greenlight: Highlighting tensorflow apis energy footprint. In *Proceedings of the 21st International Conference on Mining Software Repositories*, MSR '24, page 304–308, New York, NY, USA, 2024. ACM.

[36] Saurabhsingh Rajput and Tushar Sharma. Benchmarking Emerging Deep Learning Quantization Methods for Energy Efficiency . In *2024 IEEE 21st International Conference on Software Architecture Companion (ICSA-C)*, pages 238–242, Los Alamitos, CA, USA, June 2024. IEEE.

[37] Saurabhsingh Rajput, Tim Widmayer, Ziyuan Shang, Maria Kechagia, Federica Sarro, and Tushar Sharma. Enhancing energy-awareness in deep learning through fine-grained energy measurement. *ACM Trans. Softw. Eng. Methodol.*, 33(8), December 2024.

[38] Vijay Janapa Reddi, Christine Cheng, David Kanter, Peter Mattson, Guenther Schmuelling, Carole-Jean Wu, Brian Anderson, Maximilien Breughe, Mark Charlebois, William Chou, Ramesh Chukka, Cody Coleman, Sam Davis, Pan Deng, Greg Diamos, Jared Duke, Dave Fick, J. Scott Gardner, Itay Hubara, Sachin Idgunji, Thomas B. Jablin, Jeff Jiao, Tom St. John, Pankaj Kanwar, David Lee, Jeffery Liao, Anton Lokhmotov, Francisco Massa, Peng Meng, Paulius Micikevicius, Colin Osborne, Gennady Pekhimenko, Arun Tejusve Raghunath Rajan, Dilip Sequeira, Ashish Sirasao, Fei Sun, Hanlin Tang, Michael Thomson, Frank Wei, Ephrem Wu, Lingjie Xu, Koichi Yamada, Bing Yu, George Yuan, Aaron Zhong, Peizhao Zhang, and Yuchen Zhou. MLPerf Inference Benchmark . In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, pages 446–459, Los Alamitos, CA, USA, June 2020. IEEE.

[39] Federica Sarro. Search-Based Software Engineering in the Era of Modern Software Systems . In *2023 IEEE 31st International Requirements Engineering Conference (RE)*, pages 3–5, Los Alamitos, CA, USA, September 2023. IEEE.

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

[41] Raghavendra Selvan, Bob Pepin, Christian Igel, Gabrielle Samuel, and Erik B Dam. Pepr: Performance per resource unit as a metric to promote small-scale deep learning in medical image analysis. 2024. Presented at Northern Lights Deep Learning Conference 2025.

[42] Meghana Tedla, Shubham Kulkarni, and Karthik Vaidhyanathan. EcoMLS: A Self-Adaptation Approach for Architecting Green ML-Enabled Systems. In *2024 IEEE 21st International Conference on Software Architecture Companion (ICSA-C)*, pages 230–237, Hyderabad, India, June 2024. IEEE.

[43] Roberto Verdecchia, Luis Cruz, June Sallou, Michelle Lin, James Wickenden, and Estelle Hotellier. Data-Centric Green AI An Exploratory Empirical Study. In *2022 International Conference on ICT for Sustainability (ICT4S)*, pages 35–45, Plovdiv, Bulgaria, June 2022. IEEE.

[44] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in Software Engineering*. Springer Berlin Heidelberg, Berlin, 2024.

[45] Dustin Wright, Christian Igel, Gabrielle Samuel, and Raghavendra Selvan. Efficiency is not enough: A critical perspective of environmentally sustainable AI. 2024.

[46] Daliang Xu, Wangsong Yin, Hao Zhang, Xin Jin, Ying Zhang, Shiyun Wei, Mengwei Xu, and Xuanzhe Liu. EdgeLLM: Fast On-Device LLM Inference With Speculative Decoding. *IEEE Transactions on Mobile Computing*, 24(4):3256–3273, April 2025.

[47] Tim Yarally, Luis Cruz, Daniel Feitosa, June Sallou, and Arie van Deursen. Batching for Green AI - An Exploratory Study on Inference . In *2023 49th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 112–119, Los Alamitos, CA, USA, September 2023. IEEE.

[48] Yiyang Zhao, Yunzhuo Liu, Bo Jiang, and Tian Guo. Ce-nas: An end-to-end carbon-efficient neural architecture search framework. 2024.