ML-EcoLyzer: Quantifying the Environmental Cost of Machine Learning Inference Across Frameworks and Hardware

Anonymous submission

Abstract

Machine learning inference occurs at a massive scale, yet its environmental impact remains poorly quantified, especially on low-resource hardware. We present **ML-EcoLyzer**, a cross-framework tool for measuring the carbon, energy, thermal, and water costs of inference across CPUs, consumer GPUs, and datacenter accelerators. The tool supports both classical and modern models, applying adaptive monitoring and hardware-aware evaluation.

We introduce the *Environmental Sustainability Score (ESS)*, which quantifies the number of effective parameters served per gram of CO_2 emitted. Our evaluation covers over 1,900 inference configurations, spanning diverse model architectures, task modalities (text, vision, audio, tabular), hardware types, and precision levels. These rigorous and reliable measurements demonstrate that quantization enhances ESS, huge accelerators can be inefficient for lightweight applications, and even small models may incur significant costs when implemented suboptimally. ML-EcoLyzer sets a standard for sustainability-conscious model selection and offers an extensive empirical evaluation of environmental costs during inference.

Introduction

Concerns about the environmental impact of artificial intelligence have grown as machine learning (ML) systems are increasingly deployed in a wide range of real-world applications, from recommender systems and search engines to conversational agents and edge devices. These models now run on both resource-constrained hardware and large-scale datacenters, creating new demands for sustainable and efficient deployment strategies.

While the carbon footprint of large-scale model training has been well studied (Strubell, Ganesh, and McCallum 2019; Patterson et al. 2021), the environmental cost of inference remains underexplored—even though inference workloads often vastly outnumber training runs in practical settings (Desislavov, Martínez-Plumed, and Hernández-Orallo 2023). In production, inference is now the principal contributor to energy use and emissions in ML systems (Patterson et al. 2022).

Despite growing awareness, there is no widely adopted standard for benchmarking the environmental impact of ML inference. Existing tools and benchmarks such as MLPerf (Mattson et al. 2020) focus primarily on performance metrics (e.g., throughput, latency), and typically address only training or specific frameworks. Adaptive, cross-framework measurement protocols that cover both classical and modern models, diverse modalities, and heterogeneous hardware remain lacking.

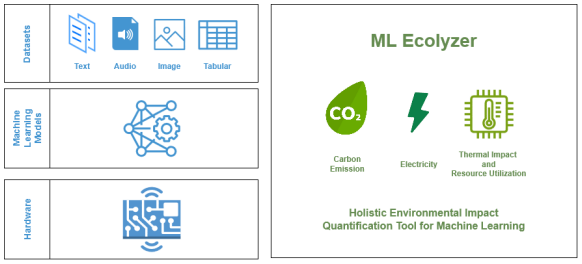

Figure 1: Overview of ML-EcoLyzer. The framework quantifies the environmental impact of machine learning inference by integrating various dataset modalities, diverse model architectures, and hardware platforms.

To address this gap, we present **ML-EcoLyzer**, an open-source, extensible tool for evaluating the environmental impact of ML inference workloads (see Figure 1). ML-EcoLyzer supports a wide array of tasks—including text, vision, audio, and tabular—and monitors real-time energy use, carbon emissions, thermal conditions, and water consumption across CPUs, consumer GPUs, and datacenter accelerators. Its adaptive monitoring engine enables consistent evaluation in both high-performance and low-resource environments. Throughout this work, we report environmental costs on a *per-inference* basis, defined as the total environmental impact (energy, carbon, water) of processing a single input sample through the complete model pipeline—for example, one text prompt-response cycle, one image classification, or one audio transcription.

We also introduce the *Environmental Sustainability Score (ESS)*, a new metric that quantifies the number of effective parameters served per gram of CO_2 emitted, enabling fair comparisons across models of different sizes, quantization levels, and operational footprints. Our empirical analysis, spanning over 1,900 inference configurations, shows that quantization substantially reduces carbon cost, and that task,

model, and hardware choices all significantly affect environmental efficiency.

The main contributions of this work are:

- We present **ML-EcoLyzer**, an open-source, framework-agnostic tool for evaluating the environmental impact of machine learning inference across both classical and modern architectures.
- We propose the *Environmental Sustainability Score (ESS)*, a parameter-normalized metric for carbon cost comparison across models, hardware platforms, and quantization settings.
- We provide a comprehensive empirical assessment covering a wide range of modalities (text, vision, audio, tabular), model families, hardware tiers, and inference precisions.

This study provides practical tools and indicators to monitor sustainability inference time, enabling actionable recommendations for resource-constrained deployments when environmental efficiency is essential.

Related Work

Carbon Footprint Analysis in Training Workloads

Early investigations of the environmental impact of machine learning have focused primarily on model training. (Strubell, Ganesh, and McCallum 2019) quantified the emissions of large-scale NLP training pipelines, estimating over 284,000 kg of CO_2 for BERT-large under hyperparameter tuning. (Patterson et al. 2021) proposed a structured emission model incorporating energy consumption (E), grid carbon intensity (CI), and datacenter efficiency (PUE):

$$\text{CO}_2 = E \times CI \times \text{PUE} \tag{1}$$

While foundational, this body of work emphasizes training cost and lacks resolution at the inference stage, where the majority of real-world energy expenditure occurs (Strubell, Ganesh, and McCallum 2019; Patterson et al. 2021; Morrison et al. 2025).

Environmental Reporting Tools and Protocols

Several tools have emerged to track and report training emissions. CodeCarbon (Courty et al. 2024) and CarbonTracker (Anthony, Kanding, and Selvan 2020) estimate carbon output using regional carbon intensity data and hardware power profiles. The Experiment Impact Tracker (Henderson et al. 2020) provides a framework for transparent emissions reporting in academic settings. However, these tools generally lack support for inference workflows, operate within narrow framework boundaries, and do not address hardware thermal or water overheads.

ML-EcoLyzer extends these capabilities by supporting inference-time profiling across a wide range of model types and hardware tiers, including CPU-only and edge-oriented deployments.

Lifecycle Benchmarks and Deployment Cost Modeling

Standardized frameworks for evaluating the environmental impact of machine learning have primarily focused on training workloads. For example, the works of (Lacoste et al. 2019) and (Morrison et al. 2025) provide systematic approaches for estimating and reporting carbon emissions, water usage, and energy consumption during model development and training. These methods represent important progress toward reproducible and transparent sustainability assessment.

However, inference workloads, which often account for the majority of operational ML activity, are not systematically benchmarked in these frameworks. Most available methodologies do not provide protocols for measuring environmental impact during the deployment phase, particularly across diverse tasks, modalities, and hardware types.

Our work addresses this limitation by introducing explicit, cross-framework, inference-time measurement protocols that cover a wide range of data modalities, model classes, and hardware configurations.

Model Compression and Inference Optimization

A variety of techniques have been proposed to improve inference efficiency, including knowledge distillation (Hinton, Vinyals, and Dean 2015), pruning (Han et al. 2015), and quantization (Jacob et al. 2018; Frantar et al. 2023). These methods reduce model size and compute requirements without significantly degrading accuracy. However, their impact on environmental metrics—such as carbon emissions and water usage—has been less systematically evaluated.

This study empirically benchmarks such techniques under consistent conditions to quantify their environmental benefits, particularly in constrained hardware scenarios.

Thermal and Water Footprint Considerations

Recent work has expanded the sustainability focus beyond carbon. (Dodge et al. 2022) identified geographic variation in datacenter water and energy intensity, and (Wu et al. 2022) called for broader environmental metrics beyond CO_2. Yet few tools incorporate these factors in a reusable benchmarking framework.

ML-EcoLyzer includes water usage estimation based on energy-to-water conversion coefficients and tracks thermal behavior in both CPU and GPU-bound workloads.

Inference Benchmarking Gaps

Significant progress has been made in standardizing the evaluation of machine learning models, especially for training efficiency and carbon emissions. For example, MLPerf (Mattson et al. 2020) and DAWNBench (Coleman et al. 2017) provide industry-wide benchmarks for speed and accuracy, while (Lacoste et al. 2019) and (Morrison et al. 2025) have introduced tools and methodologies for estimating and reporting the environmental impact of training, including holistic resource tracking.

Despite these advances, existing frameworks rarely address inference workloads in a systematic and cross-framework manner. Most prior work either focuses on training or is limited to carbon estimation without standardized protocol for measuring energy, water, or emissions during deployment. Furthermore, comprehensive benchmarking across diverse modalities (e.g., text, vision, audio, tabular), hardware platforms, and both classical and neural architectures remains largely unexplored for inference-time sustainability.

This work addresses these gaps by providing:

- Framework-agnostic support for inference evaluation on both classical and neural models,
- Inclusion of multiple environmental dimensions (energy, emissions, thermal, water), and
- A standardized benchmarking protocol adaptable to varied hardware tiers and deployment scenarios.

Together, these contributions establish a foundation for principled, reproducible environmental analysis of inference workloads in real-world settings.

ML Ecolyzer

This section formalizes the metrics used in ML-EcoLyzer for assessing the environmental impact of machine learning inference. These include carbon emissions, energy consumption, water footprint, and the proposed Environmental Sustainability Score (ESS). All metrics are reported on a *per-inference* basis—that is, the environmental cost of processing a single input sample through the complete model pipeline, enabling direct comparison across models and hardware configurations.

Carbon Emissions Estimation

Following established methodologies (Patterson et al. 2021; Courty et al. 2024), carbon emissions (in kg CO_2) are calculated using:

$$\text{CO}_2\text{ (kg)} = E\text{ (kWh)} \times CI\text{ (kg CO}_2\text{/kWh)} \times \text{PUE} \quad (2)$$

where:

- E is the total energy consumed during inference (in kWh),
- CI is the carbon intensity of the regional power grid (in kg CO_2/kWh),
- PUE denotes Power Usage Effectiveness (dimensionless), which accounts for infrastructure overhead.

For hardware classification, ML-EcoLyzer uses tier-specific PUE values based on industry benchmarks and datacenter efficiency studies (Koomey 2008; Masanet et al. 2020; Bizo et al. 2022): 1.1 for CPU-only systems (typical of edge and desktop environments with minimal cooling infrastructure), 1.2 for desktop-class GPUs (e.g., RTX, GTX series with moderate cooling requirements), and 1.4 for datacenter GPUs (e.g., A100, T4 reflecting enterprise cooling infrastructure overhead). These values align with reported data center efficiency ranges of 1.1-1.8 for modern facilities (Bizo et al. 2022).

Energy Profiling and Power Monitoring

Energy consumption (in kWh) is computed as the integral of instantaneous power over time:

$$E\text{ (kWh)} = \frac{1}{3600}\int_0^T P(t)\,dt \quad (3)$$

where $P(t)$ is instantaneous power (in watts) at time t, and T is the total duration of inference (in seconds). Power data is collected at adaptive sampling rates (typically 1–5 Hz) using system-level monitors (e.g., NVIDIA-SMI, `psutil`) and validated power models.

Water Footprint Estimation

ML-EcoLyzer estimates water usage (in liters) based on real-time power monitoring, regional water intensity factors, and device-specific cooling overhead, following the methodology established in (Lacoste et al. 2019):

$$\text{Water} = P_{\text{mon}} \times t \times WI_{\text{reg}} \times O_{\text{cool}} \times O_{\text{infra}} \quad (4)$$

where all variables are in appropriate units: P_{mon} (kW), t (h), WI_{reg} (L/kWh), and O_{cool}, O_{infra} (dimensionless). Here, P_{mon} represents power consumption estimated from real-time monitoring of CPU, GPU, and system utilization during ML workload execution, converted to energy consumption over time t. WI_{reg} represents regional water intensity coefficients from the framework's comprehensive database covering 25+ regions, ranging from 1.2 L/kWh (Iceland, geothermal/hydro) to 4.8 L/kWh (Middle East, oil/gas generation), based on the calculator methodology (Lacoste et al. 2019). O_{cool} represents device-specific cooling overhead factors (1.0× for low-power devices to 1.4× for data centers) and O_{infra} accounts for data center infrastructure water usage (1.0× to 1.2×). The framework automatically detects regional context through locale, timezone, and cloud provider environment variables, applying validated coefficients with hardware-aware overhead calculations. This monitored-energy approach provides more accurate water footprint estimates than theoretical calculations, supporting comprehensive environmental impact assessment that extends beyond carbon emissions to include actual water resource consumption (Wu et al. 2022; Patterson et al. 2021).

Effective Parameters

To enable fair comparison across models of varying size and quantization, ML-EcoLyzer introduces the notion of *Effective Parameters*. While raw parameter count reflects model capacity, it does not account for environmental differences due to quantization or precision. The effective parameter count linearly scales the total by the representational granularity:

$$\text{Effective Parameters (M)} = \frac{N \times QF}{10^6} \quad (5)$$

where N is the total number of parameters and QF is the quantization factor, representing the bit-width scaling. The default QF values are: (a) 1.0 for FP32, (b) 0.5 for FP16, and (c) 0.25 for INT8, as supported by energy and

throughput scaling in mixed-precision profiling (Micikevicius et al. 2017; Jacob et al. 2018). In models using heterogeneous precision, QF is computed as a weighted average across all layers. This linear scaling reflects the observation that lower-precision inference reduces memory, compute, and energy usage nearly proportionally, especially on hardware with native support for quantized operations (Jacob et al. 2018; Frantar et al. 2023).

Environmental Sustainability Score (ESS)

The core metric proposed in this work is the Environmental Sustainability Score (ESS), defined as:

$$\text{ESS} = \frac{\text{Effective Parameters (M)}}{\text{CO}_2 \text{ (g)}} \tag{6}$$

ESS measures how many effective parameters can be served per gram of CO_2 emitted. Higher ESS values indicate more sustainable inference configurations. The metric supports fair comparisons between full-precision and quantized models, as well as between traditional and modern architectures.

We chose parameter-based normalization over alternatives such as FLOPs-based or energy-per-token metrics for several reasons: (1) *Hardware agnosticism*: FLOPs vary dramatically by hardware implementation and compiler optimization, making cross-platform comparisons unreliable, whereas effective parameters provide a consistent measure of model capacity; (2) *Quantization awareness*: ESS inherently accounts for bit-width through the quantization factor (QF), enabling fair comparison across precision levels (FP32, FP16, INT8); (3) *Multi-modal applicability*: Unlike token-based metrics limited to sequence models, parameter normalization generalizes across text, vision, audio, and tabular domains. ESS should always be interpreted alongside absolute emissions to avoid favoring large models with high per-parameter efficiency but unsustainable total footprints (Desislavov, Martínez-Plumed, and Hernández-Orallo 2023).

Thermal Efficiency Considerations

ML-EcoLyzer captures temperature traces during inference and flags thermally inefficient regimes, particularly when GPU temperature exceeds 80°C or CPU exceeds 85°C. Such regimes incur cooling overheads, which are estimated and added to the total energy budget. These thermal adjustments follow hardware datasheet modeling and ASHRAE thermal envelope guidance.

Quantization Impact Tracking

To assess the environmental benefit of quantization, the framework tracks baseline energy and water consumption during inference execution, then applies *estimated quantization savings factors* to project multi-precision benefits. Water consumption estimates are calculated using regional water intensity factors and device-specific cooling overhead:

$$\begin{aligned} W_{\text{baseline}} = P_{\text{measured}} \times I_{\text{region}} \\ \times O_{\text{cooling}} \times O_{\text{infrastructure}} \end{aligned} \tag{7}$$

$$\begin{aligned} \text{Energy Savings (\%)} \\ \text{(estimated)} \end{aligned} = 100 \times \frac{E_{\text{measured}} - E_{\text{estimated\_quantized}}}{E_{\text{measured}}} \tag{8}$$

$$\begin{aligned} \text{Water Savings (\%)} \\ \text{(estimated)} \end{aligned} = 100 \times \frac{W_{\text{baseline}} - W_{\text{estimated\_quantized}}}{W_{\text{baseline}}} \tag{9}$$

where P_{measured} is the measured power consumption, I_{region} is the regional water intensity factor (1.2-4.8 L/kWh, (Courty et al. 2024; Lacoste et al. 2019; Micikevicius et al. 2017)), O_{cooling} and $O_{\text{infrastructure}}$ are device-specific overhead factors, and the quantized variants are estimated using predetermined savings factors. These estimates provide rapid quantization impact assessment without requiring the computational overhead of actual multi-precision inference execution. The framework supports multiple precision modes (FP32, FP16, INT8) through its quantization configuration system, enabling the potential implementation of empirical measurements. Water equivalents are provided in practical units (bottles saved, gallons conserved) alongside comprehensive regional water footprint analysis.

Experiments

We present a comprehensive evaluation of inference-time environmental costs, covering over 1,900 configurations with ML-EcoLyzer. Our study includes transformer-based LLMs, classical models, and a range of task modalities, with all results normalized to single-sample inference. This section explores how emissions, water use, and ESS are shaped by model architecture, hardware platform, task, and precision.

Model Family Analysis. Environmental impact varies by model family (Table 1). Legacy architectures such as GPT and OPT are among the most emission-intensive, with average emissions per inference of 0.121 ± 0.317 kg and 0.098 ± 0.255 kg CO_2, respectively. In contrast, newer models such as Qwen 2 (0.016 ± 0.009 kg) and Phi 3 (0.015 ± 0.002 kg) demonstrate much greater efficiency. These modern families also achieve substantially higher ESS, sometimes by an order of magnitude. This finding runs counter to the common assumption that larger or newer models are always more environmentally costly; our empirical results, as well as recent literature (Desislavov, Martínez-Plumed, and Hernández-Orallo 2023), demonstrate that architectural and precision advances can significantly reduce per-parameter emissions.

Figure 2 illustrates the dramatic environmental gap between model generations for text workloads. Despite similar parameter counts, recent releases like Qwen 2, OLMo, and LLaMA 2 deliver much lower per-minute emissions than earlier models such as GPT-2 or Pythia. This shows that relying on older, smaller models can be environmentally coun-

Model Family	CO_2 (kg, $\mu \pm \sigma$)	Water (L, $\mu \pm \sigma$)	ESS (MP/g, $\mu \pm \sigma$)
GPT	0.121 ± 0.317	1.28 ± 3.43	882 ± 4638
OPT	0.098 ± 0.255	1.05 ± 2.73	486 ± 329
Qwen 2	0.016 ± 0.009	0.14 ± 0.06	$1233 \pm 10{,}968$
Phi 3	0.015 ± 0.002	0.11 ± 0.01	585 ± 317
OLMo	0.020 ± 0.010	0.21 ± 0.11	237 ± 162
Gemma 2	0.038 ± 0.088	0.18 ± 0.66	953 ± 5749
LLaMA 2	0.029 ± 0.066	0.32 ± 0.71	678 ± 252
Playground	0.016 ± 0.001	0.17 ± 0.01	158 ± 12
HuBERT	0.0057 ± 0.0060	0.061 ± 0.062	108 ± 65
Whisper	0.0062 ± 0.0052	0.064 ± 0.053	96 ± 108

Table 1: Environmental cost by model family (per-inference basis). Metrics show average CO_2 emissions (kg), water consumption (L), and Environmental Sustainability Score (ESS: million effective parameters per gram CO_2) for processing a single input sample. Mean $\pm$ std aggregated across model variants (e.g., Qwen 1.8B, 4B, 7B) and hardware configurations. High standard deviations reflect substantial variance in hardware utilization, model sizes within families, and deployment configurations.

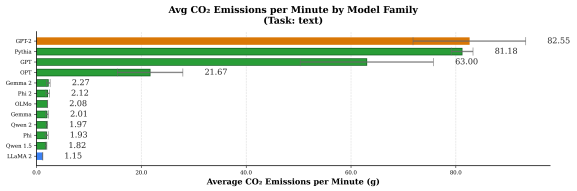

Figure 2: Average CO_2 emissions per minute by model family on text generation tasks (rate-based metric). This complements Table 1, which reports per-inference costs. Per-minute metrics reveal throughput efficiency during sustained operation, while per-inference metrics show task-level environmental cost. Error bars: standard deviation.

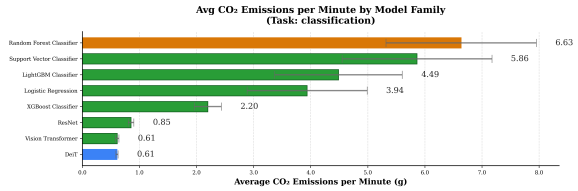

Figure 3: Average CO_2 emissions per minute by model family on classification tasks. Error bars: standard deviation.

terproductive; up-to-date architectures yield larger sustainability gains than downsizing alone.

Hardware Platform Analysis. Table 2 shows that the environmental cost of inference is highly sensitive to hardware choice. Datacenter GPUs such as the A100 deliver low average emissions (0.024 ± 0.053 kg CO_2 per run) and the highest mean ESS, but only when heavily utilized (Mattson et al. 2020). Suboptimal batching or small workloads can reduce efficiency, even on advanced accelerators, as supported by our experimental results and by findings in large-scale ML inference benchmarks (Mattson et al. 2020).

Maximizing hardware utilization through batching or workload aggregation is essential for sustainability. Deploying LLMs on large accelerators delivers high ESS only if those devices are fully used; otherwise, smaller, well-matched hardware may be more sustainable.

Task Modality Analysis. Task type also affects environmental cost. Table 3 shows that text generation (LLMs) has the highest absolute CO_2 per run, but, due to large parameter counts, still achieves high ESS. In contrast, classification and regression tasks, often dominated by traditional models implemented with scikit-learn and executed on the CPU, show extremely low ESS. This inefficiency arises because, as our measurements show (see Table 3), for small models running on general-purpose CPUs, fixed system overheads and idle resource consumption outweigh the actual computational work of inference. This is consistent with observations in recent system-level analyses (Desislavov, Martínez-Plumed, and Hernández-Orallo 2023; Barroso and Hölzle 2007). As a result, even "lightweight" pipelines can be less sustainable when deployed on CPUs without batching or hardware matching. This finding is significant for edge and resource-constrained deployments, where relying on classical ML does not necessarily translate to greener inference.

Figure 3 shows that ensemble methods like Random Forest and LightGBM are among the least efficient, often exceeding both linear and deep learning methods in per-minute emissions. Vision transformers and convolutional models, by contrast, offer lower emissions and higher throughput, especially when deployed on compatible hardware.

Quantization and Precision Effects. Many of the highest-ESS models are natively quantized or use reduced precision. As shown in Table 4, FP16 and INT8 models retain high accuracy (98.5% and 94.2%, respectively) while reducing power draw by 25% to 55% and water usage by over 0.01 L per inference. These empirical trends align with prior literature (Jacob et al. 2018; Frantar et al. 2023), confirming quantization as a dominant lever for sustainable deployment.

Most quantization gains are realized with minimal loss in accuracy for common transformer architectures. Sustainability benchmarks should always report model precision

Device Category	CO_2 (kg, $\mu \pm \sigma$)	Water (L, $\mu \pm \sigma$)	ESS (MP/g, $\mu \pm \sigma$)
Cloud (Tesla T4)	0.097 ± 0.238	0.34 ± 0.40	$991 \pm 8{,}538$
Datacenter (A100)	0.024 ± 0.053	0.02 ± 0.00	$1{,}313 \pm 9{,}128$
Consumer (RTX 4090)	0.019 ± 0.025	0.003 ± 0.003	190 ± 266
Consumer (GTX 1650)	0.047 ± 0.077	0.012 ± 0.008	164 ± 724
CPU (sklearn)	0.052 ± 0.078	0.031 ± 0.029	0.09 ± 0.47

Table 2: Environmental cost by hardware category. ESS: MP/g (mean $\pm$ std).

Task Type	CO_2 (kg, $\mu \pm \sigma$)	Water (L, $\mu \pm \sigma$)	ESS (MP/g, $\mu \pm \sigma$)
Text Generation	0.134 ± 0.269	0.30 ± 0.60	$2{,}397 \pm 13{,}400$
Classification	0.028 ± 0.042	0.03 ± 0.03	1.88 ± 7.72
Regression	0.074 ± 0.089	0.07 ± 0.09	1.18 ± 6.63
Image Generation	0.019 ± 0.006	0.02 ± 0.01	91.7 ± 52.4
Audio Processing	0.0060 ± 0.0055	0.006 ± 0.006	76.3 ± 84.9
Image Processing	0.0021 ± 0.0005	0.002 ± 0.001	35.2 ± 16.3

Table 3: Environmental cost by task type (per-inference basis). Task definitions: Text Generation (autoregressive LLMs: GPT, LLaMA, Qwen), Classification (image/text classification using ResNet, ViT, BERT), Regression (tabular prediction using scikit-learn models), Image Generation (diffusion and transformer-based synthesis: Stable Diffusion), Audio Processing (ASR and embedding: Whisper, HuBERT), Image Processing (classification and embedding extraction: DeiT, ResNet). ESS: million effective parameters per gram CO_2 (mean $\pm$ std).

and leverage available quantized releases where possible.

Monitoring Sensitivity. Measurement protocol directly affects reported emissions, especially for short or bursty inference tasks. Table 5 shows that using a 1 Hz sampling rate can overestimate emissions by nearly 6% compared to a 5 Hz standard, especially on fast models or accelerators. Adaptive monitoring strategies are necessary to ensure fair comparisons and avoid misleading results.

In general, our results highlight that sustainable ML inference depends on architectural choices, hardware task alignment, precision, and measurement practices. In the next section, we discuss the implications and recommendations.

Discussion

This study provides a systematic quantification of the environmental impact of ML inference and establishes formal benchmarks for sustainability across model families, hardware, and deployment modalities. Our results demonstrate that, when emissions are normalized per effective parameter, recent LLMs often achieve higher Environmental Sustainability Scores (ESS) than traditional ML models. This effect is driven by the ability of transformer models to amortize energy costs over a large parameter space and to utilize hardware optimized for high-throughput inference.

A key empirical finding is that traditional models, such as those implemented in scikit-learn, typically run on CPUs, where fixed system overhead and low utilization lead to very low ESS, especially in single-inference or non-batched settings. Although these classical models emit less CO_2 per run in absolute terms, their sustainability per parameter is substantially lower than that of quantized transformer models deployed on GPUs or datacenter accelerators. Hardware alignment and reduced precision significantly boost ESS for modern models.

Figure 4 summarizes the efficiency landscape for all evaluated model families. The highest efficiency region, which combines low CO_2 emissions with high ESS, is dominated by modern transformer-based models, such as the Phi, Qwen, and LLaMA series, particularly in quantized form. Classical ML models, including Random Forest, Logistic Regression, and LightGBM, consistently exhibit low ESS despite small absolute emissions. This efficiency frontier makes clear that architectural advances, hardware-software co-design, and quantization provide greater sustainability benefits than simply reducing model size.

Quantization consistently improves environmental efficiency. INT8 and FP16 models reduce per-inference emissions substantially, with negligible accuracy loss in most transformer-based tasks. These results confirm the impact of quantized deployments and support the release and adoption of lower-precision model variants.

Hardware configuration is a decisive factor in sustainability. Datacenter GPUs like the A100 deliver high ESS only under full utilization. For lightweight or edge workloads, consumer accelerators or CPUs are often more efficient. Careful alignment of workload, model, and hardware is essential for sustainable deployment.

Measurement protocol also affects results: sampling frequency and monitoring granularity have a measurable effect on emissions estimates, especially for brief or bursty inference tasks. Adaptive monitoring, as implemented in ML-EcoLyzer, ensures accuracy without introducing excess overhead.

This study establishes reliable empirical benchmarks for inference-time environmental impact and demonstrates that

Precision	Power Savings	Accuracy Retention	Water Reduction (L)
FP32	—	100%	—
FP16	25%	98.5%	0.0147
INT8	55%	94.2%	0.0152
INT4	75%	87.8%	—

Table 4: Quantization benefits: power, energy, water, and accuracy (empirical and literature-supported).

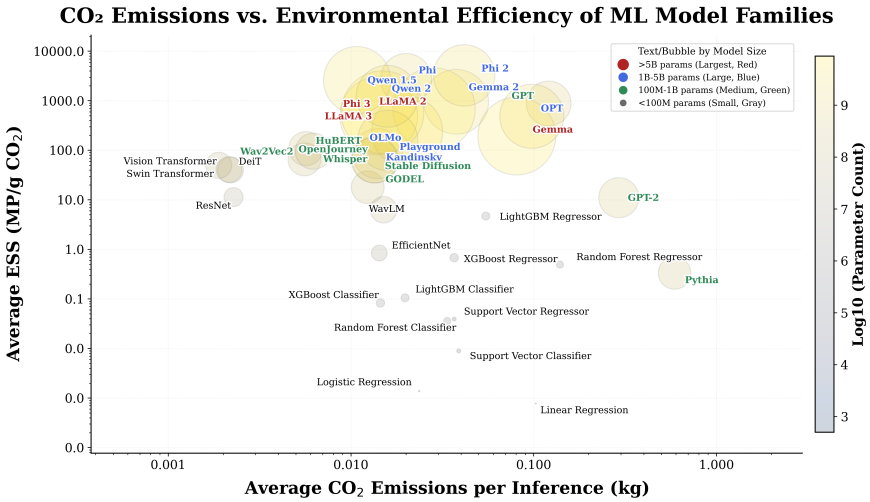

Figure 4: Sustainability Score (ESS, MP/g CO_2) vs. CO_2 emissions per inference (kg) for all model families.

Sampling Rate (Hz)	Avg CO_2 (g)	Relative Error (%)
1	27	+5.8
2	26	+1.9
5	25.5	—

Table 5: Effect of Sampling Frequency on CO_2 Estimation.

ESS serves as a robust metric for sustainability-aware machine learning deployment. It is essential that both absolute and parameter-normalized sustainability metrics, including ESS, be standardized in benchmarking and deployment studies.

Conclusion

This paper presents ML-EcoLyzer, an extensible, cross-framework, and open-source package that is designed to measure the environmental impact of machine learning inference across model types, tasks, and hardware configurations. This study emphasizes the importance of inference, a frequently overlooked but critical phase when implementing machine learning systems, underscoring the environmental costs that extend beyond emissions during the training phase.

We propose the Environmental Sustainability Score (ESS), a quantization-aware metric that captures per-parameter emissions and enables fair efficiency comparisons across models of varying precision and scale. Through over 1,900 inference runs across four hardware tiers and multiple task modalities, our benchmark provides concrete evidence for the efficiency gains of quantization, the value of hardware-utilization matching, and the role of adaptive monitoring in sustainability-aware evaluation.

ML-EcoLyzer formalizes established trends and uncovers overlooked dynamics, including the inefficiency of conventional ML models operating on idle hardware and the measurement bias that arises from coarse sampling. Releasing the tool as open-source software fosters additional experimentation and the establishment of environmental benchmarks, especially in resource-constrained or low-latency deployment scenarios. This study offers a comprehensive framework in quantifying sustainability as a core goal in the design and assessment of machine learning systems.

Potential directions for future work include expanding ML-EcoLyzer to support batched and streaming inference scenarios, adding native integration for additional ML frameworks and hardware platforms, and incorporating real-time regional grid carbon intensity data for dynamic emissions estimation. We also see value in developing more granular, task-aware sustainability metrics and in extending the analysis to encompass training workloads and multi-model serving deployments. Finally, collaborating with industry partners could help validate and refine these tools in real-world, production-scale environments.

References

Anthony, L. F. W.; Kanding, B.; and Selvan, R. 2020. Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models. ICML Workshop on Challenges in Deploying and monitoring Machine Learning Systems. ArXiv:2007.03051.

Barroso, L. A.; and Hölzle, U. 2007. The Case for Energy-Proportional Computing. *IEEE Computer*, 40(12): 33–37. Classic position paper advocating energy-proportional system design; foundational for system-level energy overhead analysis.

Bizo, D.; Ascierto, R.; Lawrence, A.; and Davis, J. 2022. Uptime Institute Global Data Center Survey. Technical Report UI Intelligence Report 51, Uptime Institute.

Coleman, C.; Narayanan, D.; Kang, D.; Zhao, T.; Zhang, J.; Nardi, L.; Bailis, P.; Olukotun, K.; Ré, C.; and Zaharia, M. 2017. DAWNBench: An End-to-End Deep Learning Benchmark and Competition. In *Advances in Neural Information Processing Systems*. End-to-end deep learning benchmark focusing on time-to-accuracy.

Courty, B.; Schmidt, V.; Goyal-Kamal; MarionCoutarel; Feld, B.; Lecourt, J.; LiamConnell; SabAmine; inimaz; supatomic; Léval, M.; Blanche, L.; Cruveiller, A.; ouminasara; Zhao, F.; Joshi, A.; Bogroff, A.; Saboni, A.; de Lavoreille, H.; Laskaris, N.; Abati, E.; Blank, D.; Wang, Z.; Catovic, A.; alencon; Stechly, M.; Bauer, C.; Lucas-Otavio; JPW; and MinervaBooks. 2024. mlco2/codecarbon: v2.4.1. Python package for tracking carbon emissions from computing.

Desislavov, R.; Martínez-Plumed, F.; and Hernández-Orallo, J. 2023. Trends in AI inference energy consumption: Beyond the performance-vs-parameter laws of deep learning. *Sustainable Computing: Informatics and Systems*, 38: 100857.

Dodge, J.; Prewitt, T.; Tachet des Combes, R.; Odmark, E.; Schwartz, R.; Strubell, E.; Luccioni, A. S.; Smith, N. A.; DeCario, N.; and Buchanan, W. 2022. Measuring the Carbon Intensity of AI in Cloud Instances. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. Analysis of carbon intensity variations across cloud providers and regions.

Frantar, E.; Ashkboos, S.; Hoefler, T.; and Alistarh, D. 2023. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. Accurate 4-bit quantization for large language models using second-order information.

Han, S.; Pool, J.; Tran, J.; and Dally, W. J. 2015. Learning both Weights and Connections for Efficient Neural Network. In *Advances in Neural Information Processing Systems (NeurIPS)*. Magnitude-based pruning achieving 9-13x parameter reduction.

Henderson, P.; Hu, J.; Romoff, J.; Brunskill, E.; Jurafsky, D.; and Pineau, J. 2020. Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research*, 21: 1–43. Experiment Impact Tracker and systematic reporting framework.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. Foundational knowledge distillation framework using temperature scaling.

Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; and Kalenichenko, D. 2018. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Fundamental quantization framework for 8-bit integer inference.

Koomey, J. 2008. Worldwide electricity used in data centers. *Environmental Research Letters*, 3(3): 034008.

Lacoste, A.; Luccioni, A.; Schmidt, V.; and Dandres, T. 2019. Quantifying the Carbon Emissions of Machine Learning. Machine Learning Emissions Calculator and systematic quantification methodology.

Masanet, E. R.; Shehabi, A.; Lei, N.; Smith, S.; and Koomey, J. 2020. Recalibrating global data center energy-use estimates. *Science*, 367(6481): 984–986.

Mattson, P.; et al. 2020. MLPerf Training Benchmark. In *Proceedings of Machine Learning and Systems (MLSys)*. Industry-standard ML training benchmark suite.

Micikevicius, P.; Narang, S.; Alben, J.; Diamos, G.; Elsen, E.; Garcia, D.; Ginsburg, B.; Houston, M.; Kuchaiev, O.; Venkatesh, G.; and Wu, H. 2017. Mixed Precision Training. *arXiv preprint arXiv:1710.03740*. FP16/BF16 training achieving 2x memory reduction and 2-6x speedup.

Morrison, J.; Na, C.; Fernandez, J.; Dettmers, T.; Strubell, E.; and Dodge, J. 2025. Holistically Evaluating the Environmental Impact of Creating Language Models. In *The Thirteenth International Conference on Learning Representations*. Holistic framework for comprehensive environmental impact assessment of language model training.

Patterson, D.; Gonzalez, J.; Holzle, U.; Le, Q.; Liang, C.; Munguia, L.-M.; Rothchild, D.; So, D. R.; Texier, M.; and Dean, J. 2022. The Carbon Footprint of Machine Learning Training Will Plateau, Then Shrink. *Computer*, 55(7): 18–28.

Patterson, D.; Gonzalez, J.; Le, Q.; Liang, C.; Munguia, L.-M.; Rothchild, D.; So, D.; Texier, M.; and Dean, J. 2021. Carbon Emissions and Large Neural Network Training. *arXiv preprint arXiv:2104.10350*. Four-factor carbon footprint model for large-scale ML training.

Strubell, E.; Ganesh, A.; and McCallum, A. 2019. Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3645–3650. Florence, Italy: Association for Computational Linguistics. Seminal paper quantifying carbon footprint of NLP model training.

Wu, C.-J.; et al. 2022. Sustainable AI: Environmental Implications, Challenges and Opportunities. *Proceedings of Machine Learning and Systems (MLSys)*. Comprehensive analysis of AI environmental impact and mitigation strategies.

Reproducibility Checklist

Instructions for Authors:

This document outlines key aspects for assessing reproducibility. Please provide your input by editing this `.tex` file directly.

For each question (that applies), replace the "Type your response here" text with your answer.

Example: If a question appears as

```
\question{Proofs of all novel claims
are included} {(yes/partial/no)}
Type your response here
```

you would change it to:

```
\question{Proofs of all novel claims
are included} {(yes/partial/no)}
yes
```

Please make sure to:

- Replace ONLY the "Type your response here" text and nothing else.
- Use one of the options listed for that question (e.g., **yes**, **no**, **partial**, or **NA**).
- **Not** modify any other part of the `\question` command or any other lines in this document.

You can `\input` this .tex file right before `\end{document}` of your main file or compile it as a stand-alone document. Check the instructions on your conference's website to see if you will be asked to provide this checklist with your paper or separately.

---

1. General Paper Structure

1.1. Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes/partial/no/NA) yes.

1.2. Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (yes/no) yes.

1.3. Provides well-marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper (yes/no) yes.

2. Theoretical Contributions

2.1. Does this paper make theoretical contributions? (yes/no) yes.

If yes, please address the following points:

2.2. All assumptions and restrictions are stated clearly and formally (yes/partial/no) yes.

2.3. All novel claims are stated formally (e.g., in theorem statements) (yes/partial/no) no. not applicable.

2.4. Proofs of all novel claims are included (yes/partial/no) no. not applicable.

2.5. Proof sketches or intuitions are given for complex and/or novel results (yes/partial/no) no. not applicable

2.6. Appropriate citations to theoretical tools used are given (yes/partial/no) yes.

2.7. All theoretical claims are demonstrated empirically to hold (yes/partial/no/NA) NA

2.8. All experimental code used to eliminate or disprove claims is included (yes/no/NA) yes

3. Dataset Usage

3.1. Does this paper rely on one or more datasets? (yes/no) yes.

If yes, please address the following points:

3.2. A motivation is given for why the experiments are conducted on the selected datasets (yes/partial/no/NA) yes

3.3. All novel datasets introduced in this paper are included in a data appendix (yes/partial/no/NA) NA.

3.4. All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no/NA) NA.

3.5. All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations (yes/no/NA) ues

3.6. All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available (yes/partial/no/NA) yes.

3.7. All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing (yes/partial/no/NA) NA.

4. Computational Experiments

4.1. Does this paper include computational experiments? (yes/no) yes.

If yes, please address the following points:

4.2. This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting (yes/partial/no/NA) yes

4.3. Any code required for pre-processing data is included in the appendix (yes/partial/no) no. not applicable.

4.4. All source code required for conducting and analyzing the experiments is included in a code appendix (yes/partial/no) no, however the codes are hosted in open source repository.

4.5. All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no) yes.

4.6. All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes/partial/no) yes.

4.7. If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results (yes/partial/no/NA) yes

4.8. This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks (yes/partial/no) yes.

4.9. This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics (yes/partial/no) yes.

4.10. This paper states the number of algorithm runs used to compute each reported result (yes/no) yes.

4.11. Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information (yes/no) yes.

4.12. The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank) (yes/partial/no) no. not applicable

4.13. This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments (yes/partial/no/NA) no. not applicable.