

EcoCompute AI: A High-Fidelity Framework for Measuring and Optimizing Carbon Footprint in Large-Scale AI Training

Hongping Zhang

<https://github.com/hongping-zh/ecocompute-ai>

Independent Researcher / Open Source Contributor

January 26, 2026

Abstract

As Large Language Models (LLMs) continue to scale, the environmental impact and operational costs associated with their training have become significant barriers to sustainable AI development. Current carbon tracking tools often rely on static Thermal Design Power (TDP) values, leading to substantial estimation errors. This report introduces **EcoCompute AI**, an open-source framework designed to bridge the gap between theoretical estimation and real-world energy consumption. By implementing **L3 Hardware Grounding**—aligning dynamic telemetry with MLPerf benchmarks and regional grid carbon intensity—EcoCompute AI provides a high-fidelity audit of training costs and emissions. We demonstrate the utility of the framework through a CI/CD-integrated monitoring system and an interactive web-based estimator.

1 Introduction

The computational demand for training state-of-the-art AI models is doubling approximately every six months. The industry-wide shift toward LLMs has led to a surge in energy consumption, yet the tools available for developers to monitor these impacts remain fragmented. Most existing solutions offer only “black-box” estimates that fail to account for hardware-specific efficiency or workload variations.

EcoCompute AI aims to democratize “Green AI” by providing developers with actionable insights directly within their existing workflows (e.g., GitHub Actions). Our focus is on transparency, accuracy, and ease of integration.

2 Methodology: L3 Hardware Grounding

The core innovation of EcoCompute AI is its hierarchical energy estimation model, specifically the **Level 3 (L3) Grounding** approach. We define three tiers of accuracy:

- **L1 (Static Estimations):** Uses hardware manufacturer TDP specs. Highly inaccurate as it ignores idle power and dynamic utilization.
- **L2 (Dynamic Telemetry):** Captures real-time power draw via APIs like `nvidia-smi`. While better, it lacks context regarding the efficiency of the specific ML kernel being executed.
- **L3 (Grounded Telemetry):** Calibrates dynamic power data against **MLPerf training baselines**. It adjusts for architectural overheads and identifies inefficiencies in the software-hardware interface.

2.1 Mathematical Framework

The total energy consumption (E_{total}) is modeled as:

$$E_{total} = \left(\int_0^T (P_{static} + \alpha \cdot P_{dynamic}(u)) dt \right) \times PUE \quad (1)$$

Where:

- P_{static} is the baseline power of the node.
- α is the **Calibration Coefficient** derived from specific hardware-workload pairings (e.g., Transformer-based models on NVIDIA H100).
- u represents real-time utilization.
- PUE (Power Usage Effectiveness) is the data center efficiency factor, dynamically retrieved based on the cloud region.

3 System Architecture

EcoCompute AI is built as a modular ecosystem:

1. **Core Engine:** A Python library that interfaces with system drivers to capture fine-grained energy metrics.
2. **CI/CD Integration:** A GitHub Action that post-processes training logs to generate a *Sustainability Report*.
3. **EcoCompute Calculator:** A frontend tool that allows for pre-training cost estimation.

4 Case Study: LLaMA-7B Training Simulation

Using our L3 Grounding model, we simulated the training of a LLaMA-style 7B parameter model (Table 1).

Table 1: Estimated Costs for LLaMA-7B Training

Parameter	Value
Hardware	64 × NVIDIA H100 SXM
Data Volume	2.0 Trillion Tokens
Cloud Region	EU North (Sweden)
Estimated Cost	\$82.5K
Carbon Footprint	330 kg CO₂e

The analysis identified that integrating *FlashAttention-2* could potentially reduce training time by 2-4x.

5 Conclusion

EcoCompute AI provides a necessary bridge between machine learning engineering and environmental sustainability.