

How Expensive Is AI Coding? Measuring Energy and Emissions

L. Casella and T. N. Vartziotis
TWT GmbH Science & Innovation

Tuesday, October 6th, 2026

Aachen Automotive Colloquium 2026



CONTENT

1. TWT SCIENCE & INNOVATION
2. MOTIVATION
3. SUSTAINABLE SOFTWARE
4. AI INFERENCE & ENERGY ESTIMATION
5. DEMONSTRATOR
6. LIFE CYCLE ASSESSMENT



Dr. Laura Casella
Customer Relations Manager



Tina Vartziotis, Ph.D candidate NTUA / Harvard
Research & Academic Relations

TWT SCIENCE &
INNOVATION

01

| WE TRANSFER KNOWLEDGE – INTO MARKET-READY TECHNOLOGIES |

TECHNICAL AND SCIENTIFIC TRANSFER

AI & Data, Simulation, Processes, Quantum Computing

Our Commitment: Best Performance at the Best Price

Highly qualified professionals

300

from 17 nations – academic,
interdisciplinary, excellent

Partner organizations

100+

from leading universities, global industry
players, and our trusted partner in Greece

Self-financed, very strong credit rating

100%

independent – for over 25 years

Research Projects

100+

innovative, application-oriented, and
future-oriented

Years as a technology pioneer

40

in technical and scientific transfer

Professors

20+

former TWT staff now in academia and
research

MANAGEMENT: Dr. Dimitris Vartziotis, Dipl.-Ing. Frank Beutenmüller, Dr. Michael Keckeisen.

Image Source: Stock Images



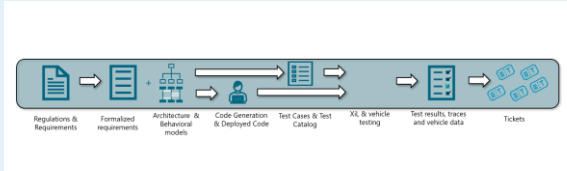
WE KNOW HOW.



TRANSFORMATION: AI & DATA, SIMULATION, PROCESSES, QUANTUM

PIONEER PROJECTS: INNOVATION, SPEED, QUALITY, COST REDUCTION, COMPETITIVE ADVANTAGE

AI-Based (SW-)Development



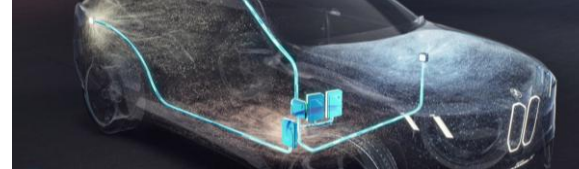
with FORMALIS®

AI-Based Physics Simulation



with OPTIMALIS®

AI @ Test & Defect Mgmt.



with DETECTIS®

AI-Based Interior Morphing



with MORPHIS®

AI-Based Resource Mgmt.



with DELPHIS®

AI-Based Flying Fuel Cell



AI-Based Electric Power Net

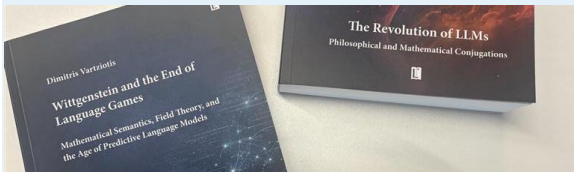


with DYNAMIS® and BATSIS®

AI-Based Patent Process



Trusted LLMs



with GSFT®

Trusted AI-Physics



with KANs

Engineer's Digital Twin



with AI-Specialist®

Enterprise AI-Orchestration & Cost Optimization

300-500 Mio. EUR	60-80 %	55-165 Mio. EUR	< 12 Monate
------------------	---------	-----------------	-------------

| WE TRANSFER KNOWLEDGE – INTO MARKET-READY TECHNOLOGIES |

BASED ON SCIENCE & OPERATIONAL EXCELLENCE

Formal methods enhanced AI-Agents – innovation through international partnerships and expert network.



TISAX L3
DIN ISO

TWT





50+
countries
research activities



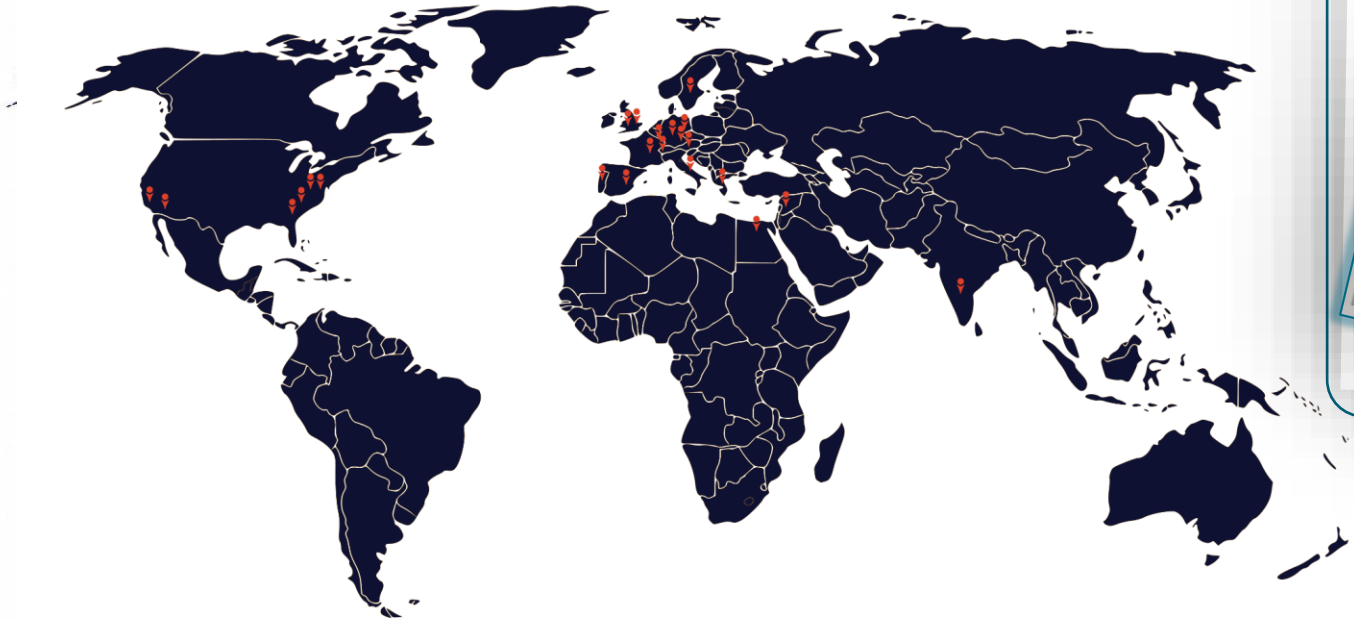
200+
academic partners
worldwide



50+
leading institutions
in our network



Regular
publications &
joint research outputs
with Mercedes-Benz



A global network of excellence

Connecting top minds to solve tomorrow's mobility challenges.

Development projects with:



AMG

DAIMLER TRUCK

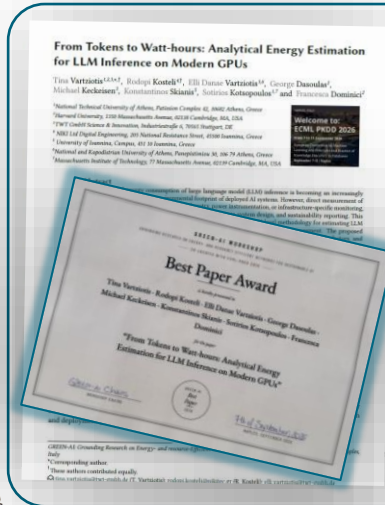


PORSCHE

CARIAD



Selected Publications & Publicly funded Research



2026

From Tokens to Watt-hours: Analytical Energy Estimation for LLM Inference on Modern GPUs

National Technical University of Athens
Massachusetts Institute of Technology
Harvard University
TWT GmbH Science & Innovation

European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases - ECML PKDD 2026 7.-11.Sept, Neapel

- Petrovic, N. et al.: GenAI for Automotive Software Development: From Requirements to Wheels, arXiv:2507.18223, 2025
- P. Vaudrevange et al., "Kolmogorov-Arnold Networks in Industrial Application of Truck Development," in Stuttgart International Symposium on Automotive and Powertrain Technology 2026
- Shift2SdV (2025 - 2028), Modular Middleware & API Framework for automotive applications
- ReKonIK (2026 - 2029), Resiliente Kollaborationsstrukturen durch KI-generierte Modelle,

PARTNERING WITH LEADING INSTITUTIONS WORLDWIDE



Massachusetts
Institute of
Technology



Stanford
University



UNIVERSITY OF
OXFORD



Technische
Universität
München

ETH zürich

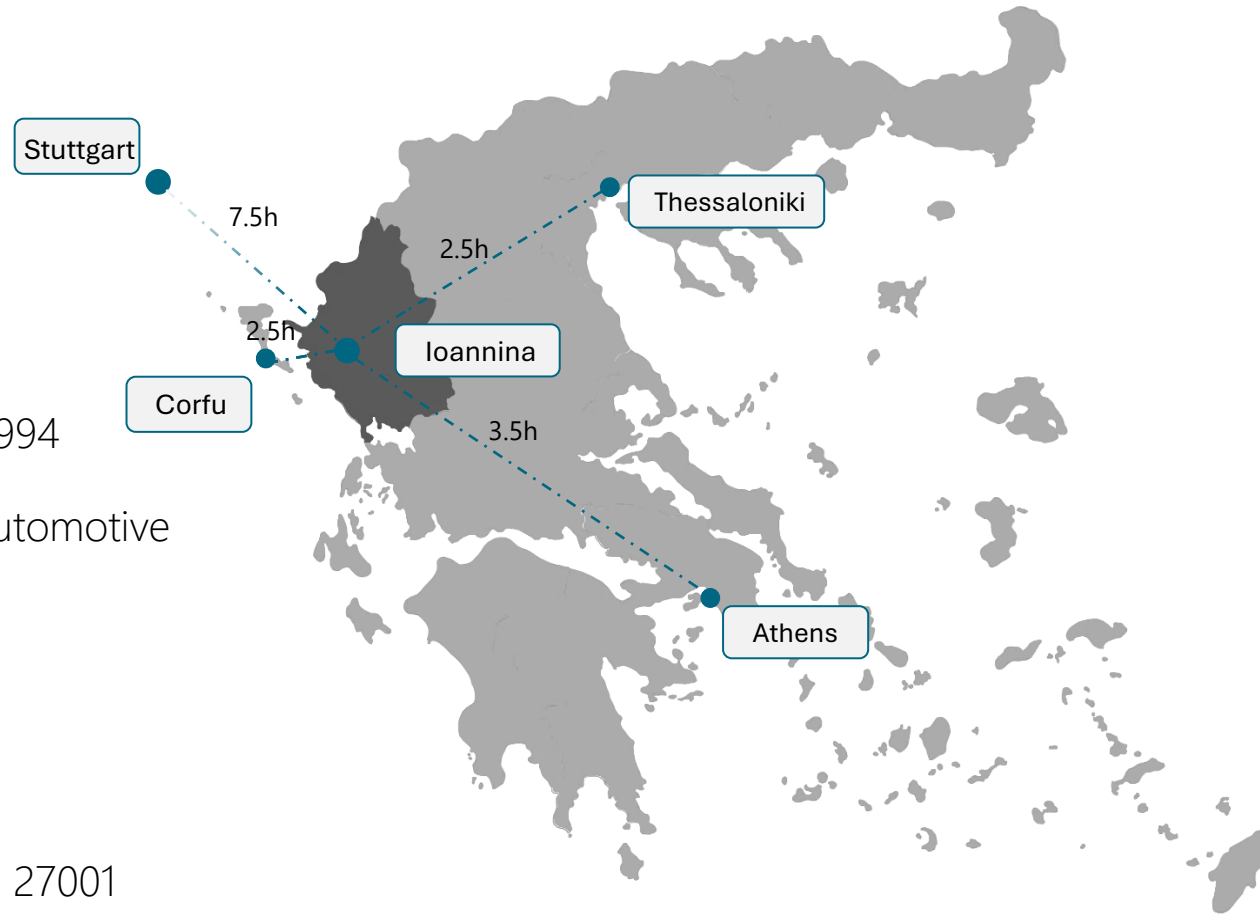
Berkeley
UNIVERSITY OF CALIFORNIA

and more
partners
worldwide

YOUR ADVANTAGE – OUR REGIONAL PARTNER NIKI / COOPERATION PARTNER



- ✓ Ioannina, Greece ·
CEO: Dr. D. Vartziotis
- ✓ IT & Digital Engineering · since 1994
- ✓ Projects for TWT at · Premium automotive
OEMs
- ✓ European research activities
- ✓ Approx. 60 employees
- ✓ Certifications TISAX, 9001, 14001, 27001
- ✓ Project languages · German and English



30+ years partnership

High number of talents
from technical Universities

Silicon Valley of Greece

Mutual visits to promote
collaboration

Easy accessibility

MOTIVATION

02

TWT @ GREENCODE RESEARCH PROJECT



Goals TWT

- New methods for green coding in industrial applications and R&D
- Exploring or setting energy standards
- Combining green coding with AI / SLM / LLM



Partners



a.o.



Bundesministerium
für Bildung
und Forschung



Massachusetts
Institute of
Technology



National
Technical
University of
Athens

JEVONS PARADOX - WHY EFFICIENCY IS NOT ENOUGH

What is the Jevons Paradox?

Originally observed in the 19th century in relation to coal use: more efficient steam engines did not necessarily reduce coal consumption because they made coal-powered applications cheaper and more widespread.

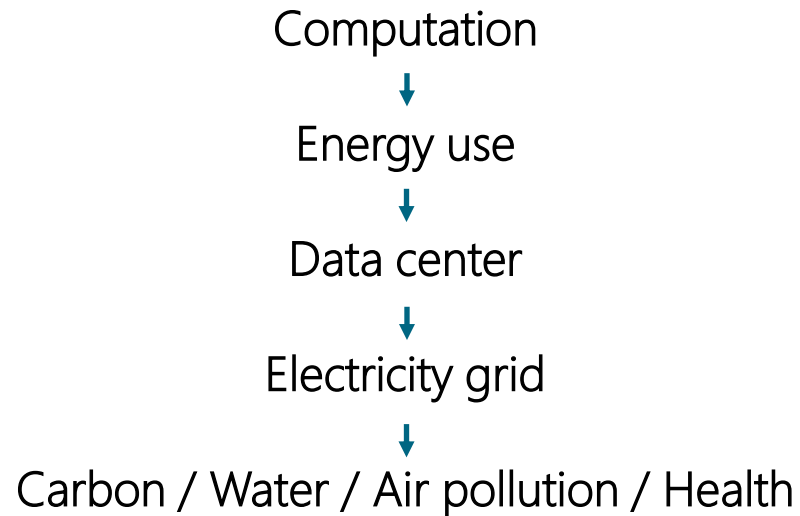
Efficiency improvements can reduce the resources required per unit of activity, while simultaneously increasing total demand enough to offset — or even exceed — those savings.

Why it matters for AI?

- **More efficient models:** fewer Wh per inference
- **Cheaper AI services:** more users & more requests
- **More capable systems:** larger models & longer contexts & new applications

Sustainable computing cannot be evaluated only through efficiency; it must also account for scale, demand growth, and total system-level impacts.

RESEARCH GAP- WE CANNOT **MANAGE** WHAT WE CANNOT MEASURE



Current limitations

- Energy use is rarely directly observable.
- Different AI models have very different computational characteristics.
- Data-center reporting is often highly aggregated.
- Environmental impacts depend strongly on location and time.
- Corporate sustainability claims do not always map to physical system impacts.

There is a disconnect between computational activity and its real-world environmental consequences.

Motivation: Connecting Software, AI and Resource Use

Why does sustainable computing matter?



AI growth



More computation



More infrastructure



More environmental pressure



1. Scale
AI workloads
are rapidly
increasing
computational
demand.

2. Physical infrastructure
Digital computation ultimately requires:

Processors
Servers
Cooling
data centers
electricity and water



3. Environmental consequences
Energy consumption translates into
impacts that depend on:



Where
When
How

computation takes place.

*The digital world is **not** immaterial. Every computation is ultimately a **physical** process.*

-   Sustainable Software
-   AI Inference & Energy Estimation
-   Data Centers & Physical Infrastructure
-   Life Cycle Assessment of Software Development

SUSTAINABLE
SOFTWARE

04

Sustainable Software

Learn to Code Sustainably: An Empirical Study on LLM-based Green Code Generation

Tina Vartziotis^{1,2}, Ippolyti Dellatolas³, George Dasoulas⁴, Maximilian Schmidt¹, Florian Schneider¹, Tim Hoffmann⁵, Sotirios Kotsopoulos^{2,3}, Michael Keckeisen¹

¹TWT GubH Science & Innovation
²National Technical University of Athens
³Massachusetts Institute of Technology
⁴Harvard University
⁵Mercedes-Benz

Abstract.—The increasing use of information technology has led to a significant share of energy consumption and carbon emissions from data centers. These contributions are expected to rise with the growing demand for big data analytics, increasing digitization, and the development of large artificial intelligence (AI) models. The need to address the environmental impact of software development has led to increased interest in green (sustainable) coding and claims that the use of AI models can lead to energy efficiency gains. Here, we provide an empirical study on green code and an overview of green coding practices, as well as metrics used to quantify the sustainability awareness of AI models. In this framework, we evaluate the sustainability of auto-generated code. The auto-generated codes considered in this study are produced by generative commercial AI language models, GitHub Copilot, OpenAI ChatGPT-3, and Amazon CodeWhisperer. Within our methodology, in order to quantify the sustainability awareness of these AI models, we propose a definition of the code's "green capacity", based on certain sustainability metrics. We compare the performance and green capacity of human-generated code and code generated by the three AI language models in response to easy-to-hard problem statements. Our findings shed light on the current capacity of AI models to contribute to sustainable software development.

1. INTRODUCTION

A. Environmental Impact of AI

In the era of rapid technological advancement, the environmental impact of software development has become a pressing concern. Advances in software and hardware techniques have led to a significant increase in the demand for digital services, with internet traffic increasing 25-fold since 2010 [1]. Global data center electricity consumption, excluding cryptocurrency mining, was estimated to be 240-340 TWh in 2022, representing 1-1.3% of global electricity consumption and 0.3% of global carbon emissions [1]-[3]. Countries with expanding data center markets are seeing rapid data center electricity consumption growth. For example, data center electricity consumption in Ireland accounted for 20% of the country's total electricity use in 2022, triple its 2015 levels [4]. Energy efficiency gains have helped limit data center energy demand, including improvements in IT hardware, cooling systems, and a shift from local processors to large-scale datacenters; yet increased computing demand for applications such as blockchains and machine learning could outpace efficiency gains. Artificial

intelligence and machine learning are areas where the demand is expected to grow. A 2018 OpenAI study found that the amount of compute (in 10^{15} floating point operations per second - day) has been increasing exponentially since 2012 due to algorithmic innovation and the development of new AI models [5]. The lifecycle emissions of an AI model encompass the emissions generated during the development of the model (training and testing phase), as well as those resulting from running the model once it is deployed (inference phase or queries). Several studies have estimated the training cost of AI models, including natural language processing (NLP) models that are highly computationally expensive [6]-[8]. Estimated CO₂ emissions from training a large NLP model are 1-10 times those from the lifecycle of a car [6]. The energy consumption of large NLP models during training ranged from 20MWh to more than 1200MWh, with the development of GPT-3 estimated to have generated 552 tons of CO₂ equivalent (GCO_{2e}) [7]. These emissions only encompass the training phase of the models. Once the models are deployed, their usage triggers additional energy demand due to consumer queries. The energy consumption of the most cutting-edge computer vision and NLP models during inference was found to have increased exponentially [9], with total emissions depending on the number of consumer queries, or how frequently the model is used (known as the multiplicative factor). Recent generative AI language models have already experienced increased demand, triggering large numbers of queries since their release, amounting to an energy consumption in the inference phase that is a significant contributor to the model lifecycle emissions [10]. For example, assuming 100 million daily queries - estimated from data from SimilarWeb - and an energy use per query of roughly 0.002kWh - based on the average energy required by a forward pass of an NLP model [9] - the energy consumption of ChatGPT can be estimated to be about 0.2GWh per day.

Despite the significant carbon footprint associated with the development and use (training and inference) of AI models, they are said to have the potential to contribute to environmental sustainability. This potential is encapsulated in the "carbon handprint" of software, where the use of the software leads to energy savings [8]. One application of generative AI models

Can software itself be more sustainable?

② Research question

Can AI-generated code be optimized not only for correctness, but also for computational efficiency?



Approach

Compare alternative implementations based on:

- execution time
- resource utilization
- energy-related performance



Finding

Functionally equivalent code can have substantially different computational costs.

$$GC(\mathcal{M}, \mathcal{T}) = \sum_{i \in \mathcal{P}} \max \left(\text{PD} \left(X_i^{(P_{\text{init}})}, X_i^{(i)}, c^{(i)} \right), 0 \right)$$

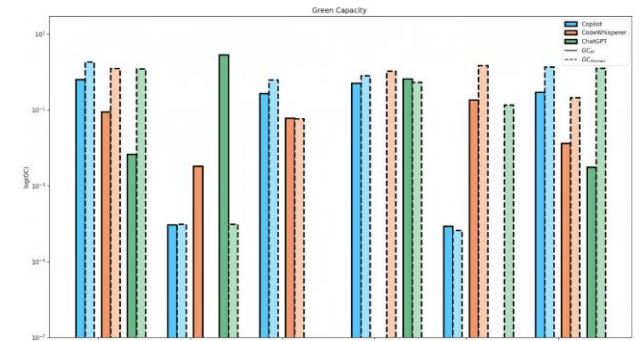


Fig. 2. Comparison plot of Green Capacity values among the three automated code generation models (Copilot, CodeWhisperer, ChatGPT) throughout six coding problems (3Sum, Cookies, Median, Network, Search, Sort). For each reported GC_{Ai} value (depicted with the full line bar), we append on its right the computed GC_{Human}, that compares the human submission with the initially generated code for each model. A higher left value implies that the code generation model performed better than the human submission throughout the sustainability metrics. Absence of data denotes a value of 0.

LLM4Code Workshop, ICSE, 2024

<https://doi.org/10.48550/arXiv.2403.03344>

Sustainability begins before the hardware is even selected.

EXISTING TOOLS OVERVIEW

Tool matrix (1-10)

Power/energy consumption, efficiency and environmental-impact coverage by tool

Tools matrix

Tool / metric	Power / energy	Efficiency	Environmental impact	Hardware / scope	Software / flow
EMRCO	Yes	Yes	Yes	CPU, GPU, full mode via wallmeter and RAPL	CIVICO GILab integration
CodeCarbon	Yes	Yes	Yes	CPU/GPU/RAM/Cloud	Python Package
Green Cost Explorer	Yes	Yes	Yes	AWS Regions	Cost + Carbon Overlay
Kube Green	Yes	Yes	Yes	Kubernetes Nodes	Kubernetes Add-on
Javalix	Yes	Yes	Yes	Java Applications	Java Agent
Kapler	Yes	Yes	Yes	Kubernetes Nodes	K8P + Prometheus
Website Carbon Calculator	Yes	No	Raises awareness	Web Hosting Servers	Front-end/Website Analyzer
Customer Carbon Footprint Tool	Yes	Yes	Yes	AWS	Cloud Usage Analyzer
Emissions Impact Dashboard	Yes	Yes	Yes	Azure/Microsoft Cloud	Dashboard Integration
Cloud Carbon Footprint	Yes	Yes	Yes	AWS, GCP, Azure	Open source CLI & Dashboard

13

Tool matrix (11-20)

Power/energy consumption, efficiency and environmental-impact coverage by tool

Tools matrix

Tool / metric	Power / energy	Efficiency	Environmental impact	Hardware / scope	Software / flow
Green Metrics Tool	Yes	Yes	Yes	CPU/Software Systems	Software Lifecycle Analyzer
Carbon Footprint (reporting tool)	Yes	Yes	Yes	Cloud Providers	Reporting Dashboard
EcoCode	No	Yes	Yes	Android Devices	Static Analyzer for Android
RedBint	No	Yes	Yes	Web Browsers	Static Code Analysis
ProSP	Yes	Yes	Yes	Full IT Stack	Automated Evaluation
Power Distribution Unit (PDU)	Yes	No	No	Anything with external power supply	---
Watts Up Pro	Yes	No	No	Anything with external power supply	---
IPM	Yes	No	No	Server board	---
Intel RAPL	Yes	No	No	CPU and Memory energy consumption	---
nvidia-smi	Yes	No	No	Nvidia GPU	---

23

Tool matrix (21-30)

Power/energy consumption, efficiency and environmental-impact coverage by tool

Tools matrix

Tool / metric	Power / energy	Efficiency	Environmental impact	Hardware / scope	Software / flow
PowerTop	Yes	No	No	General purpose	---
Automater	Yes	No	No	General purpose	---
yMeter	Yes	---	---	General purpose	---
GreenDroid	Yes	Yes	Yes	Android Devices	Android Test Suite
GreenHub	Yes	No	Yes	Android Devices	Community Power Profiling
Carbon Tracker	Yes	Yes	Yes	GPU/CPU	DL Training Monitor
OSP Green Maturity Matrix	No	Yes	Yes	Organizational Tools	Self-Assessment Framework
GCPI (GreenCode-PowerMeter)	Yes	Yes	No	Any CPU (model-based, host-independent)	VS Code extension
PowerNode	Yes	No	No	Servers, Data Store, Switches (Cloud)	GreenWave Gateway
eco-ci-energy-estimation	Yes	Yes	Yes	CPU	Integrated into CI pipeline

33

AI INFERENCE & ENERGY ESTIMATION

05

THE CONTINUOUS COST OF AUTOREGRESSIVE GENERATION

Growing AI-assisted development

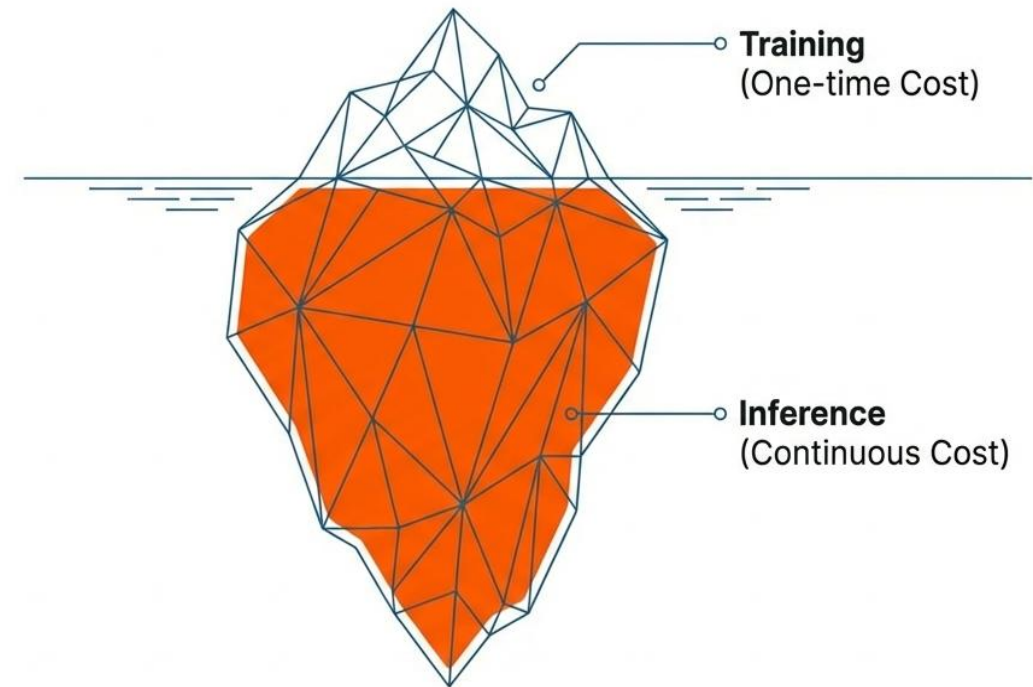
- AI programming and vibe coding market projected to grow from \$4.7B to \$12.3B by 2027 [[Taskade, 2026](#)]
- ~30% of new code at Google and Microsoft is AI-generated [[Second talent, 2026](#)]
- 25% of YC W25 startups have codebases that are ~95% AI-generated [Fawzy et al., 2026]

Lifetime energy footprint

- Depends on generated tokens, sequence length, model size, hardware configuration and batching

Deployment impact

- Applies equally to on-premise/small-to-medium GPU deployments



THE HARDWARE TELEMETRY BOTTLENECK

The problem: Direct energy measurement is hardware-bound and non-portable.

Hardware telemetry & power instrumentation [A. Lacoste et al., L. Lannelongue et al.]

- Direct access to GPU telemetry or power sensors

Coarse-grained carbon-accounting tools (SPROUT) [B. Li et al.]

- High-level reporting, no request- or token-level estimates

Transformer scaling-law & GPU benchmarks [J. Kaplan et al., J. Hoffmann et al., O. Antepara et al.]

- Basis for analytical FLOP & hardware-energy modeling, no request-level estimates

Our approach: An analytically structured, empirically calibrated GPU-level methodology without runtime measurement.



AI Inference & Energy Estimation

From Tokens to Watt-hours: Analytical Energy Estimation for LLM Inference on Modern GPUs

Tina Vartziotis^{1,2,*}, Rodopi Kosteli^{4,†}, Elli Danae Vartziotis^{4,‡}, George Dasoulas², Michael Keckeisen³, Konstantinos Skianis⁵, Sotirios Kotsopoulos^{4,‡} and Francesca Dominici²

¹National Technical University of Athens, Patission Complex 42, 10702 Athens, Greece

²Harvard University, 150 Massachusetts Avenue, 02138 Cambridge, MA, USA

³TWT GmbH Science & Innovation, Industriestrasse 4, 70631 Stuttgart, DE

⁴NIKI Ltd Digital Engineering, 201 National Resistance Street, 45100 Ioannina, Greece

⁵University of Ioannina, Campus, 451 10 Ioannina, Greece

[†]National and Kapodistrian University of Athens, Panepistimiou 30, 107 79 Athens, Greece

[‡]Massachusetts Institute of Technology, 77 Massachusetts Avenue, 02139 Cambridge, MA, USA

Abstract

The operational energy consumption of large language model (LLM) inference is becoming an increasingly important component of the environmental footprint of deployed AI systems. However, direct measurement of inference energy often requires hardware telemetry, power instrumentation, or infrastructure-specific monitoring, limiting its applicability in comparative studies, early-stage system design, and sustainability reporting. This report presents an analytically structured, empirically calibrated, GPU-level methodology for estimating LLM inference energy on NVIDIA H100-class accelerators without direct runtime measurement. The proposed estimator combines parameter-scaled transformer FLOP accounting, calibrated memory-traffic factors, and hardware-specific energy coefficients for FP16/BF16 tensor-core computation and high-bandwidth memory movement. It explicitly separates prompt prefill from autoregressive decoding, enabling energy estimates for input tokens, output tokens, and complete inference requests. The methodology further decomposes total energy into compute, parameter access, key-value-cache write, and attention read components, allowing the scaling behavior with model size, context length, and generated token count to be analyzed. The resulting estimates are not intended to replace physical power measurements; rather, they provide transparent, reproducible, and assumption-explicit approximations suitable for model comparison, green-coding analysis, and design-time evaluation of LLM inference workloads.

Keywords

Large Language Models, GPU Inference, Inference Energy Modeling, Transformer Systems, Computing Emissions, Tensor Core Computing, Energy-Efficient AI, Green AI, Sustainable Computing, High-Performance Computing

1. Introduction

As large language models (LLMs) scale to serve millions of users across cloud, edge, and on-premise deployments, their cumulative inference energy has emerged as a first-order sustainability concern, driven by the substantial computational and environmental costs of large-scale AI inference. Prior work in Green AI has shown that advances in model capability are often accompanied by increases in energy consumption and carbon emissions, motivating energy-aware approaches to machine learning research and deployment [1, 2].

While early work focused primarily on training costs [3], broader Green AI research highlighted the importance of computational efficiency, energy-aware reporting, and environmental impact across the ML lifecycle [2, 4, 5, 6], including recent efforts to extend energy-aware optimization to model selection [7]. Attention has since shifted toward inference, which is becoming an increasingly important operational component of deployed LLM systems.

Unlike training, which is performed occasionally, inference workloads run continuously and scale with user demand through autoregressive token generation, accumulating substantial energy costs over a model's deployment lifetime [8]. Their energy footprint depends strongly on generated-token count, sequence length, model size, hardware configuration, and batching behavior [9, 10, 11].

This makes inference energy a practical concern not only for large cloud providers, but also for applied AI teams, research groups, and small-to-medium organizations deploying LLMs on local or on-premise GPU infrastructure. For such deployments, accelerator-side energy does not capture the full service-level cost, since system and facility overheads, including cooling, networking, and power delivery, also contribute [12, 13]. Nevertheless, GPU-side energy is often a major controllable component of inference cost in GPU-dense systems [14], making GPU-level estimation useful for comparing model and workload choices, assessing green-coding interventions [15], and making deployment-time decisions when full datacenter-level instrumentation is not available.

Existing approaches for estimating or reporting AI-related energy consumption often rely on hardware telemetry, external power measurement, infrastructure-specific monitoring, or coarse-grained carbon-accounting tools [6, 5]. Measurement-based studies provide valuable empirical evidence across hardware configurations and task types [16, 17], but their results are tied to specific hardware, serving systems, and runtime configurations. Carbon-aware inference work such as SPARK further demonstrates that autoregressive generation can itself be optimized for sustainability [18]. Conversely, transformer scaling-law studies and GPU microarchitectural benchmarks provide the

*Corresponding author.

[†]These authors contributed equally.

✉ Tina Vartziotis@twt-gmbh.de (T. Vartziotis);
rodopi.kosteli@niki.gr (R. Kosteli); elli.vartziotis@twt-gmbh.de
(E. D. Vartziotis); gdomici@mit.edu (F. Dominici);
michael.keckeisen@twt-gmbh.de (M. Keckeisen); kskianis@uoi.gr
(K. Skianis); skots@mit.edu (S. Kotsopoulos);
fdomici@mit.edu (F. Dominici);
✉ 0000-0002-0877-7063 (T. Vartziotis); 0000-0001-7116-9138
(R. Kosteli); 0009-0006-1309-8719 (E. D. Vartziotis);
0000-0002-0562-5136 (G. Dasoulas); 0000-0002-9421-8566 (K. Skianis);
0000-0002-9421-8566 (S. Kotsopoulos); 0000-0002-9421-8566
(F. Dominici);
© 2022 Copyright for this paper by its authors. Use granted under Creative Commons License
Attribution 4.0 International (CC BY 4.0).

How much energy does an AI inference require?

② Research question

Energy measurement is hardware-bound. How can we measure it?

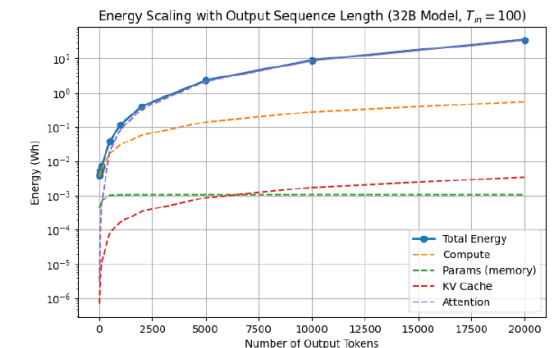
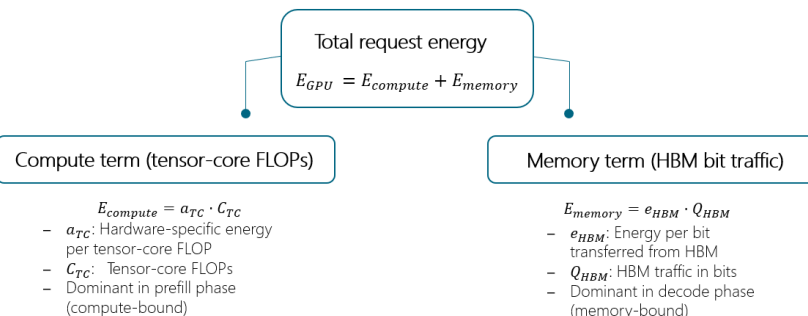
🔄 Approach

We estimate how AI workloads translate into energy use by considering:

- the amount of computation required
- the number of input and output tokens
- the size of the model
- the length of the context

🔍 Finding

A hardware-aware analytical model that estimates inference energy without requiring direct power measurements for every workload.



GreenAI Workshop, ECML-PKDD, 2026

<https://doi.org/10.48550/arXiv.2607.26571>

A token represents a piece of text, not a fixed amount of energy.

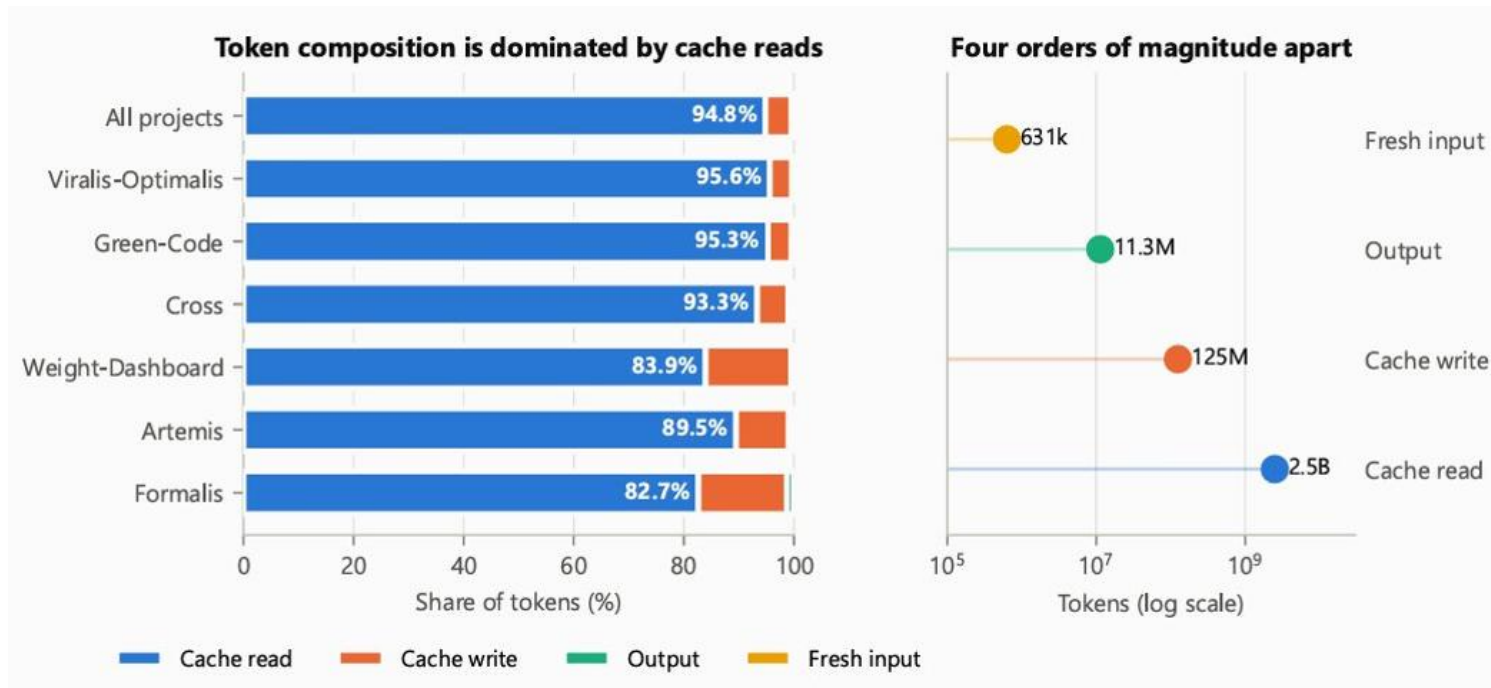
WORKLOAD ANALYSIS – TOKEN DISTRIBUTION

Input tokens — tokens in the prompt/context that haven't been cached; processed at full input rate.

Output tokens — tokens autoregressively generated by the model.

Cache write — the first pass over a context segment, where its KV-cache state gets persisted for reuse; priced at a premium over standard input (reflects the cost of computing and storing that state).

Cache read — a cache hit: reusing previously computed KV-cache state instead of recomputing it; priced at roughly 10% of standard input rate.



Standard energy models that only measure output or input tokens miss over 90% of the actual workload; modeling KV-cache memory reads is essential for accurate energy estimation.

EMISSIONS OF DIFFERENT SOFTWARE ENGINEERING TASKS

Heavyweight Tasks: Refactoring & Testing

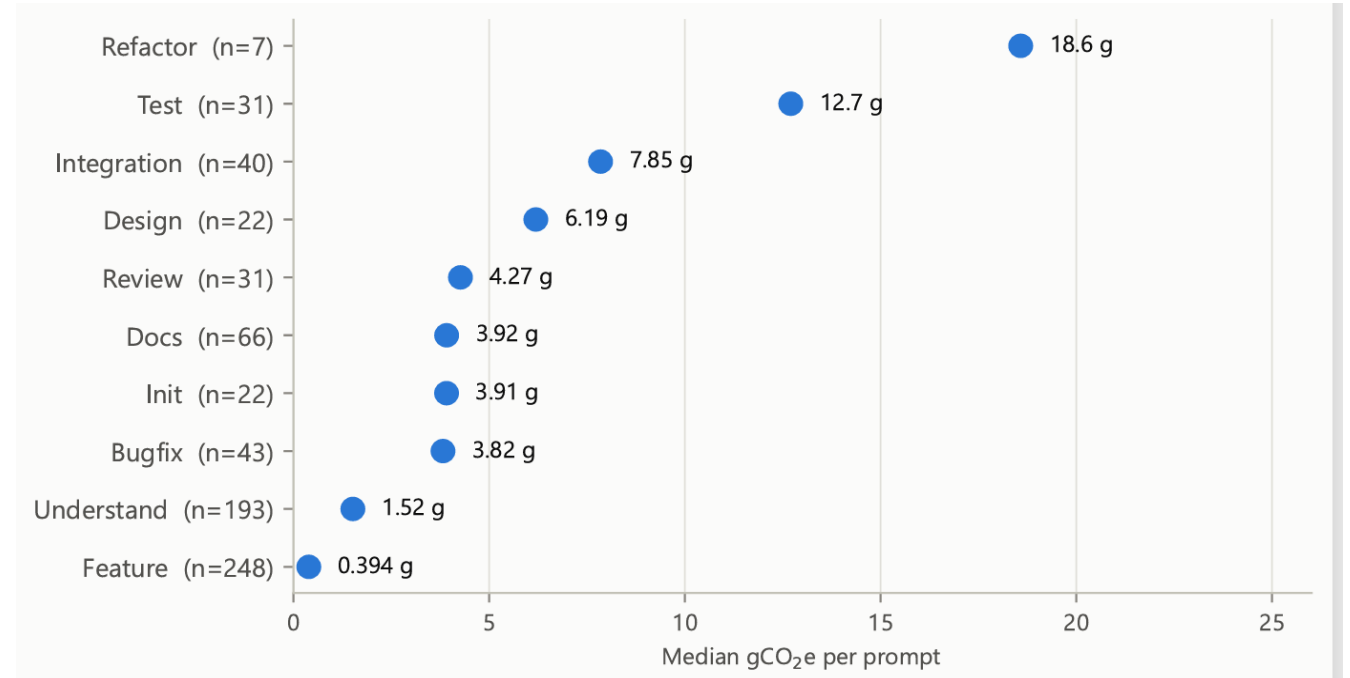
- Reconstructing complex legacy code requires heavy context window & multi-step reasoning

High Frequency vs. Low Footprint

- Feature and Understand prompt are the most frequent, but they emit very little carbon per prompt

47x Impact Disparity & Optimization Target

- A simple refactor prompt consumes as much as 47 Feature prompts



Feature — building new functionality that didn't exist before

Understand — reading/exploring code to answer a question, no changes made

Bugfix — fixing broken or incorrect existing behavior

Init — initial project setup or onboarding (e.g. /init-generated CLAUDE.md, first-time environment setup)

Docs — writing or updating documentation, comments, or docstrings

Review — reviewing existing code or a diff for correctness/quality, no new feature work

Design — planning architecture or approach before writing code

Integration — connecting separate systems or components together

Test — writing, fixing, or running tests

Refactor — restructuring existing code without changing what it does

THE HYBRID FRAMEWORK: EXTENDS ECOLOGITS WITH OUR ANALYTICAL METHODOLOGY

EcoLogits [[Samuel Rincé et al.](#)]

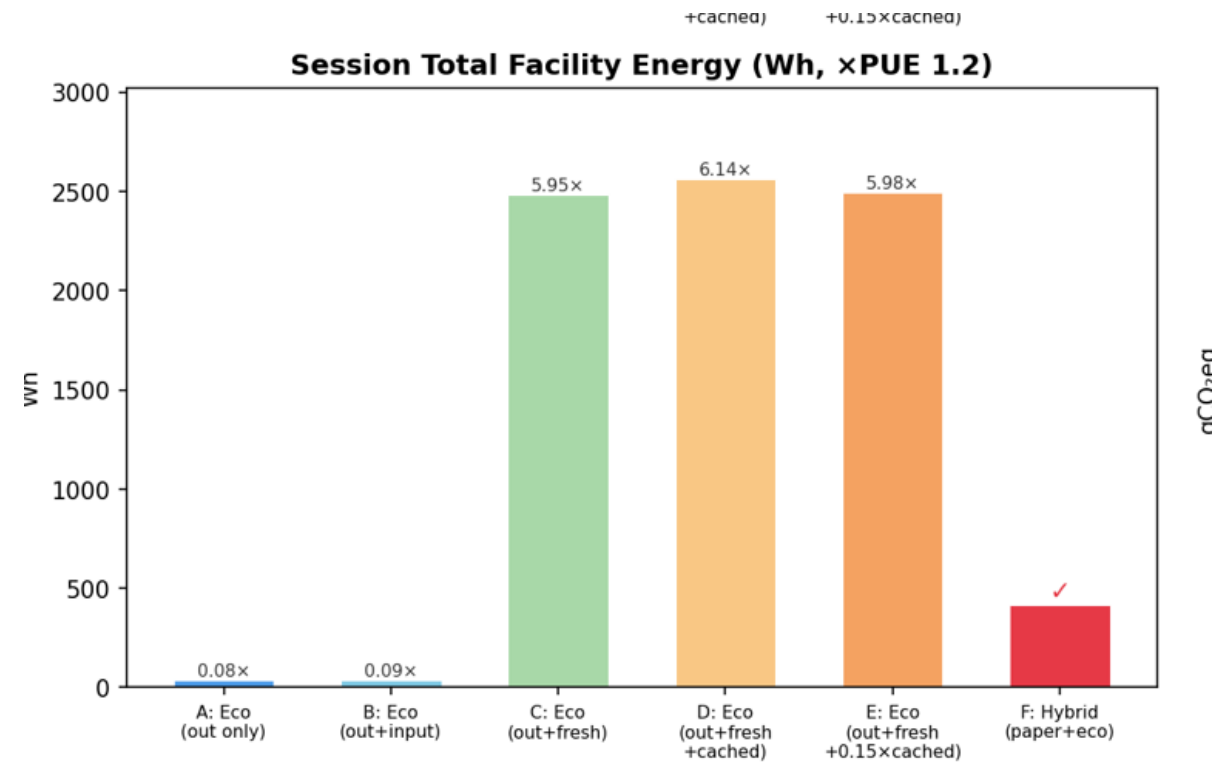
- An open-source tool from CodeCarbon (python library / web calculator / API for estimating the environmental footprint of generative-AI inference)
- Output tokens-only model assumption

Agentic Memory Load

- Over 94% of token volume consists of KV-cache memory reads
- Output generation accounts for less than 1% of total GPU data movement

Hybrid Framework

- Replaces output-token regression with an analytical hardware model [[Vartziotis et al.](#)]
- Applies the infrastructure parameters used by the EcoLogits tool (Datacenter PUE, Grid Carbon Intensity, embodied emissions)



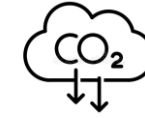
LIFE CYCLE ASSESSMENT

06

LIFE CYCLE ASSESSMENT (LCA) IN GREENCODE



LCA
environmental
impact across a
product's full life
cycle.



Software's carbon footprint isn't obvious. It lives in the servers it runs on, the laptops it's built with, and the AI tools queried during development.



Our goal: produce a single GWP-100 score (kg CO₂-eq) for a real software product, combining electricity, hardware, cloud services, and waste into one comparable number.

LCA ANALYSIS PIPELINE

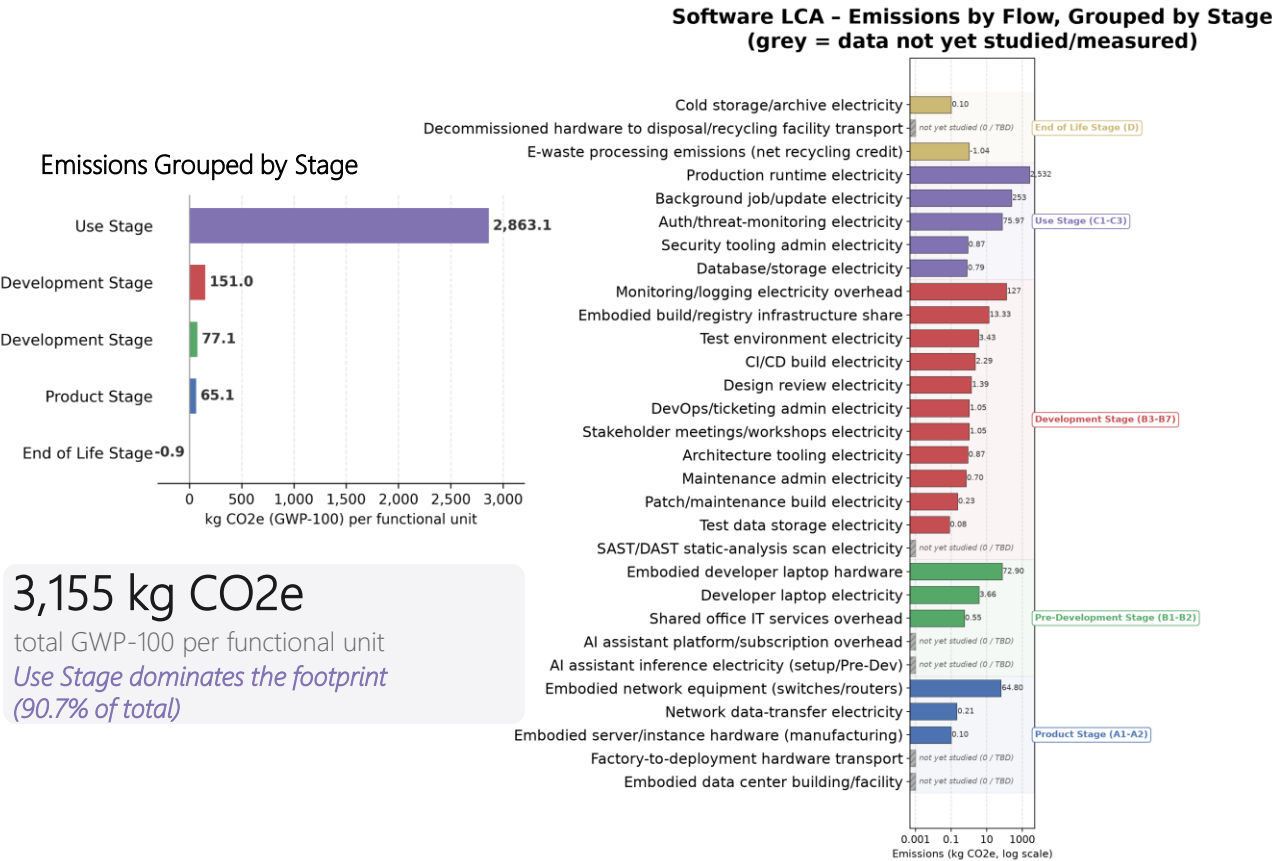
Our Contribution

-  Custom LCI Dataset
34 process flows mapped across all five ISO 14040 life-cycle stages, built specifically for software systems.
-  Data-Quality Review
Every flow assessed and tagged: verified, pending review, excluded, or flagged for methodological uncertainty.
-  Automated Assessment Pipeline
Python scripts import our dataset into Brightway2, apply GWP-100 characterization, and generate stage-level results.

Example: our LCI dataset (excerpt)

Process	Flow	Amount	Unit
Application Execution	Production runtime electricity	10,464	kWh
Development Environment	Developer laptop electricity	15.12	kWh
Hardware Infrastructure	Embodied server hardware	0.1	kg CO2e

Our Results : Emissions Grouped by Stage / Emissions grouped by Flow



LIST OF PUBLICATIONS

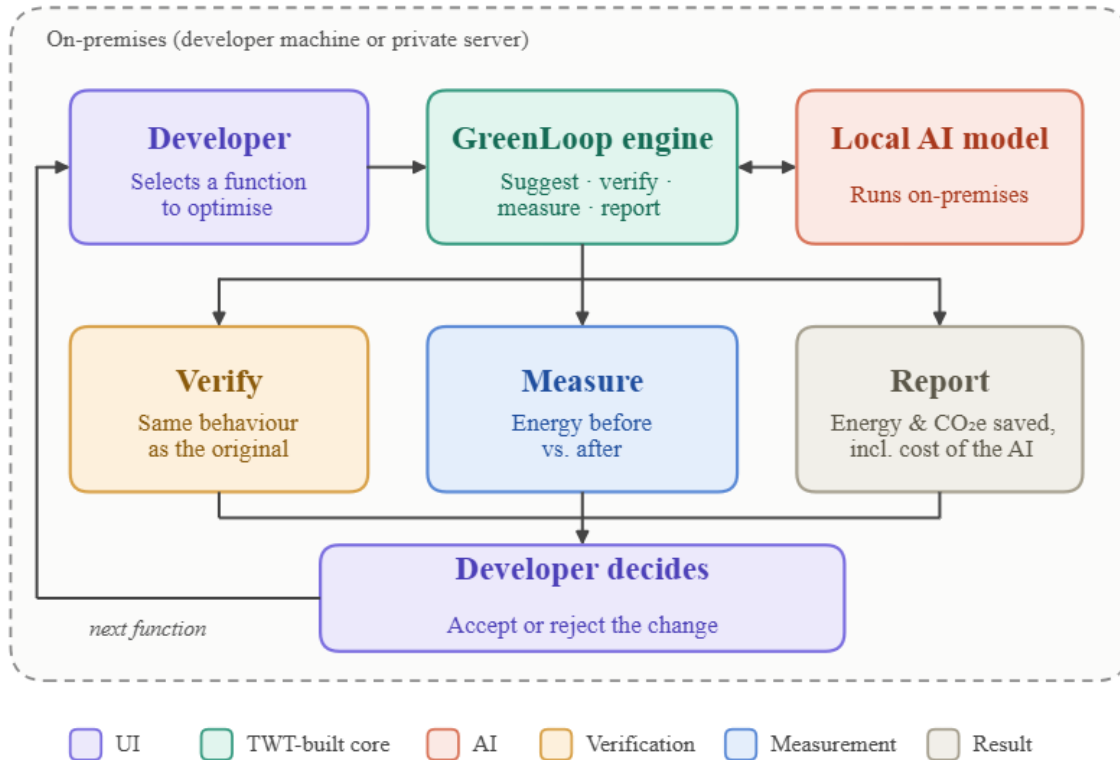
1. Vartziotis, T., Kosteli, R., Vartziotis, E., Dasoulas, G., Keckeisen, M., Skianis, K., Kotsopoulos, S., & Dominici, F. (2026). *From Tokens To Watt-hours: Analytical Energy Estimation For Llm Inference On Modern Gpus*. Arxiv:2607.26571.
2. Vartziotis, L., Vartziotis, T., Beutenmueller, F., Salta, S., Moraitis, K., Katsaros, M., & Kotsopoulos, S. (2026). *Emotive Architectures: The Role Of Llms In Adjusting Work Environments*. Arxiv:2604.25601.
3. Dominici, F., & Vartziotis, T. (2025). Integrating Environmental Health In The Ai Era: An Urgent Imperative. *Harvard Data Science Review*, 7(4). <https://doi.org/10.1162/99608f92.7d27980c>
4. Vartziotis, E., Vartziotis, T., Dasoulas, G., Dellatolas, I., Dominici, F., Kotsopoulos, S., & Tympas, A. (2025). *A Systematic Analysis Of Ai Trends In Environmental Research*. Research Square. <https://doi.org/10.21203/rs.3.rs-7138335/v1>
5. Vartziotis, T., Schmidt, M., Dasoulas, G., Dellatolas, I., Attademo, S., Le, V. D., Wiechmann, A., Hoffmann, T., Keckeisen, M., & Kotsopoulos, S. (2024). Carbon Footprint Evaluation Of Code Generation Through Llm As A Service. In *2024 Stuttgart International Symposium On Automotive And Engine Technology* (Pp. 230–241). Springer. https://doi.org/10.1007/978-3-658-45010-6_15
6. Vartziotis, T., Dellatolas, I., Dasoulas, G., Schmidt, M., Schneider, F., Hoffmann, T., Kotsopoulos, S., & Keckeisen, M. (2024). Learn To Code Sustainably: An Empirical Study On Llm-based Green Code Generation. In *Proceedings Of The 1st International Workshop On Large Language Models For Code (Llm4code '24)* (Pp. 30–37). <https://doi.org/10.1145/3643795.3648394>

DEMONSTRATOR

03

TWT DEMONSTRATOR- GREENLOOP

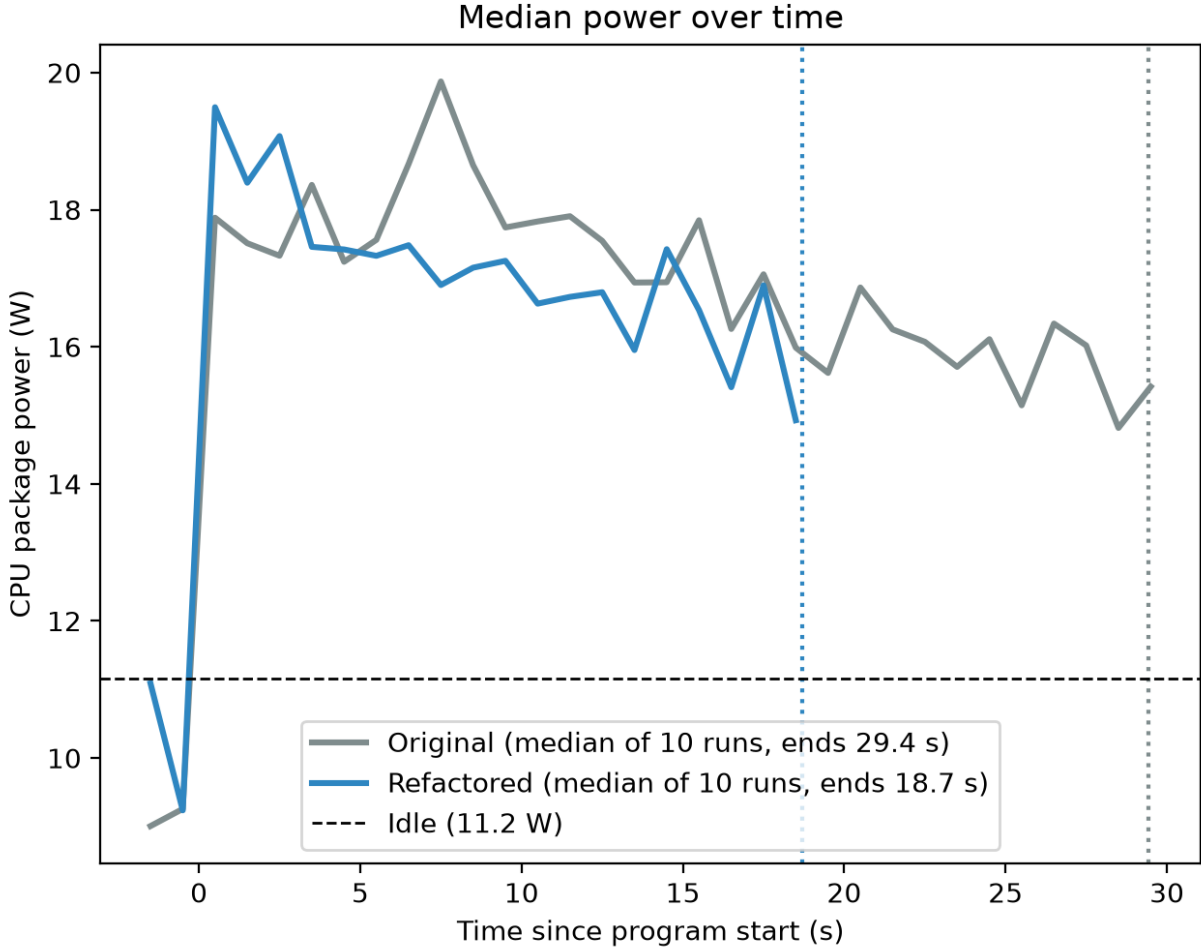
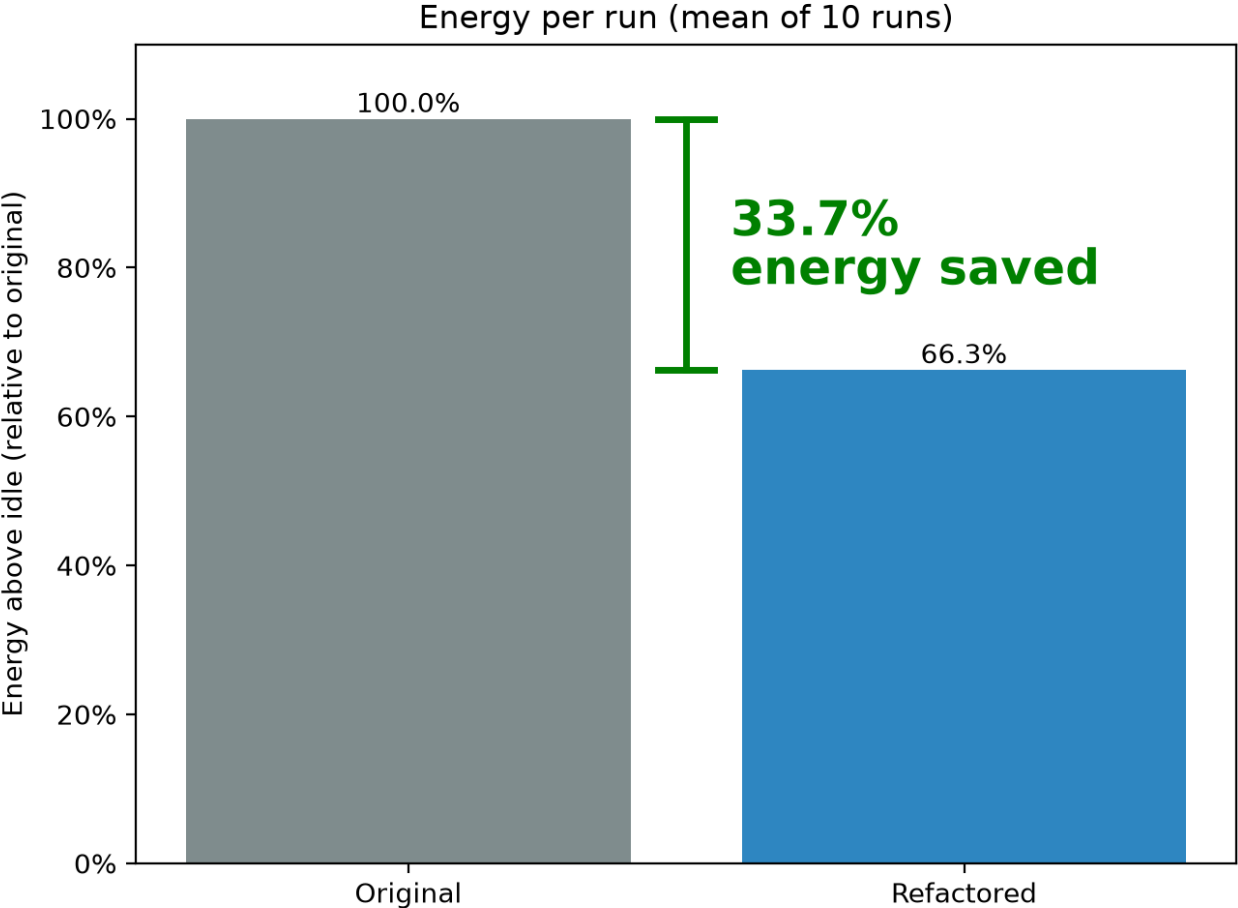
GreenLoop – a closed loop on your own infrastructure
All components run locally – no code or data leaves the company



A closed loop on the developer's machine:

1. Runs on-premises, so your code never leaves your infrastructure
2. Every change must behave exactly like the original
3. Energy is measured before and after, including the cost of the AI

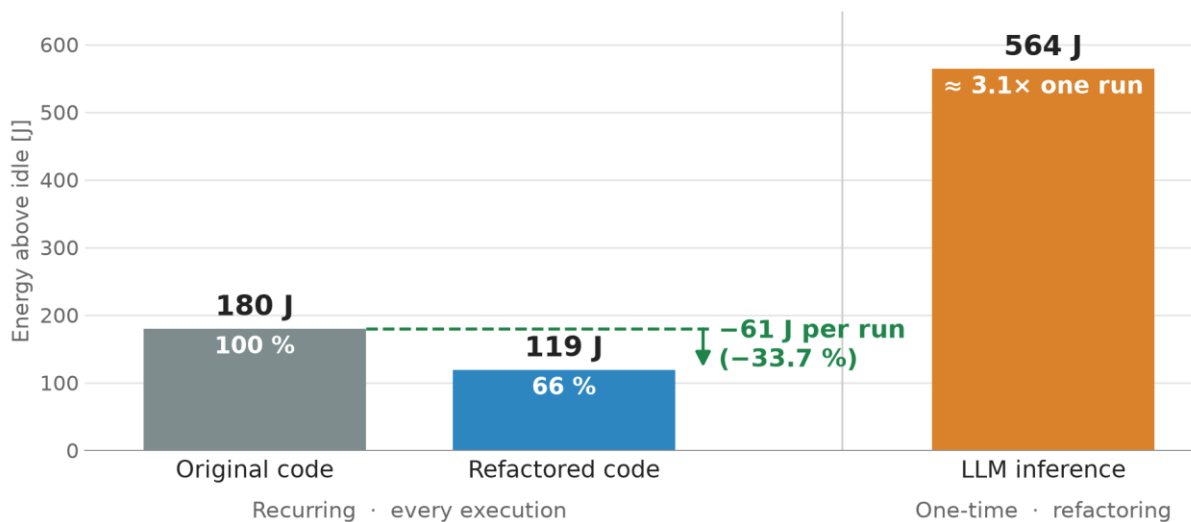
AstarSearch::expandNodes - Original vs Refactored



DEMONSTRATOR OUTCOME + INFERENCE COST

The LLM refactoring pays for itself after ~10 runs

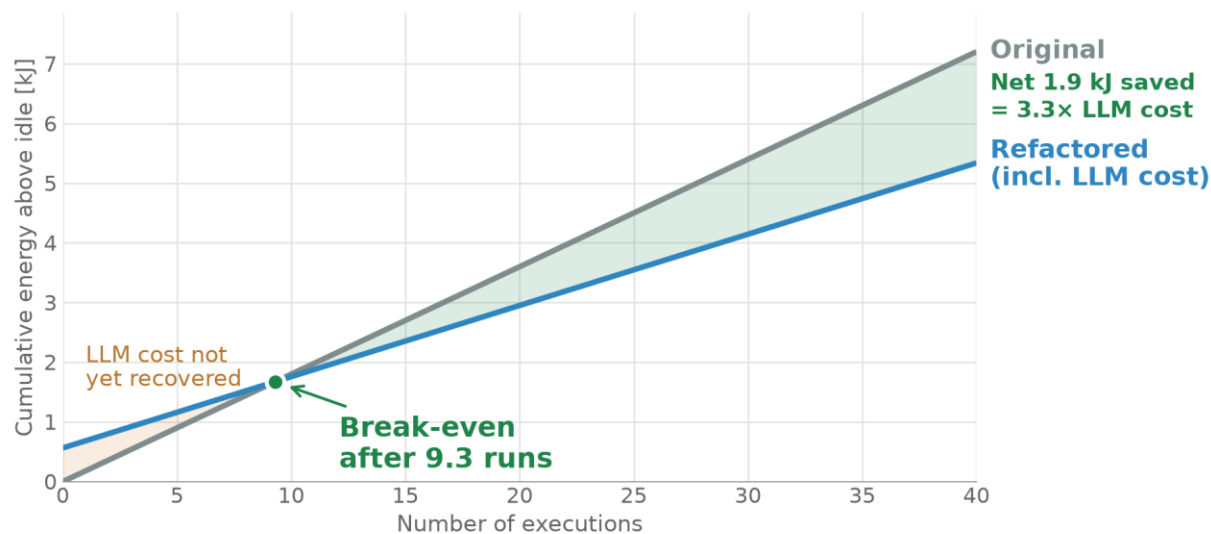
AstarSearch::expandNodes — energy per run (mean of 10 runs) vs. one-time LLM inference



LLM cost 564 J ÷ 60.7 J saved per run = 9.3 runs to break even. CPU package energy above idle (idle baseline 11.2 W).

Every further run is a permanent 33.7 % saving

AstarSearch::expandNodes — cumulative energy incl. one-time LLM inference (564 J)



CONTACT PERSONS



Dr. Laura Casella

laura.casella@twt-gmbh.de



Tina Vartziotis

tina.vartziotis@twt-gmbh.de



TWT
WE KNOW HOW.

