Transparent Screening for LLM Inference and Training Impacts

Arnault Pachot Thierry Petit

March 14, 2026

Abstract

This paper presents the transparent screening framework used in ImpactLLM to estimate inference and training impacts for current large language models. It documents the algorithms that convert natural-language application descriptions into bounded environmental estimates, and it also reports the comparative tables and visual results published in the observatory. The objective is not to claim direct measurement for opaque proprietary services, but to provide an auditable, source-linked proxy framework that improves comparability and public discussion under conditions of limited observability.

Keywords: LLMs, source-linked evidence, screening method, natural language interface.

1 Introduction

This paper documents the transparent screening framework used in the *ImpactLLM Observatory*. The observatory now covers 40 models, and this manuscript explains how the interface extracts scenarios from natural-language descriptions, propagates them through bounded multi-factor screening proxies, and publishes the resulting comparative tables and figures. The goal is to keep those proxies auditable while keeping the application workflow fast and interpretable.

The remainder of this paper covers the technical design of the inference estimator, the training proxy, the comparative outputs currently exposed in the observatory, and the limits of those assumptions.

The methodological premise is straightforward. For the dominant hosted LLM services, direct provider-side environmental telemetry is generally unavailable, even though those systems dominate practical use and public debate. In that context, avoiding approximation altogether does not produce a more rigorous discussion; it leaves room for opaque claims, unsupported comparisons, and contradictory numerical narratives. The role of the present paper is therefore not to claim direct measurement where none exists, but to define a bounded, source-linked, inspectable proxy framework that is methodologically preferable to unverifiable assertions.

2 Inference Screening Method

The main practical question addressed by the tool is straightforward: given a software feature using an LLM, can we produce an inference estimate that is useful for screening and comparison when direct provider telemetry is absent?

The current answer is deliberately limited. The estimator is *inference-only*. It excludes model training, embodied impacts, application-side software consumption, and ancillary infrastructure from the displayed result. This is not because those dimensions are unimportant, but because the available inference anchors are structured enough to support a transparent screening method, whereas mixing them with broader lifecycle terms would blur the meaning of the result.

The design objective is also practical speed. Instead of asking users to fill a large technical form, the web interface starts from a natural-language description such as "We use GPT-4o-mini for customer support, around 4,000 uses per month in France." A parser then maps this text to a compact scenario with model, request type, approximate tokens, usage volume, and country assumptions. The output is therefore fast to obtain, but every inferred parameter remains visible to the user and can be inspected or challenged.

The current market-model release follows five rules.

- It starts from an observed literature anchor rather than from provider claims. The retained prompt-energy anchor is the Gemini Apps median prompt energy reported by Elsworth et al.: 0.24 Wh/prompt [8].

- It makes token volume explicit. Prompt compute is approximated from input and output tokens, with a larger weight for output generation than for input processing.
- It does not scale on raw parameters alone. It constructs *effective active parameters* by adjusting the target profile with context-window class, serving mode, modality support, and architecture notes.
- It reports a bounded low-central-high interval rather than a single falsely precise point.
- It derives carbon from retained energy and the electricity mix of the retained country, which is a contextual assumption rather than a model measurement.

Formally, let $E_a = 0.24$ Wh be the anchor value and $P_a = 180$ the active-parameter proxy retained for the anchor model. For a target model t, let P_t denote active parameters, T_{in} input tokens, and T_{out} output tokens. We define weighted prompt-compute volume:

$$V_t = T_{in} + \omega T_{out}, \tag{1}$$

with $\omega = 1.8$ in the current release. The project reference scenario is:

$$V_{ref} = 1000 + 1.8 \times 550 = 1990. \tag{2}$$

Effective active parameters are then defined by:

$$P_{t,s}^{eff} = P_t \times F_{t,s}^{ctx} \times F_{t,s}^{srv} \times F_{t,s}^{mod} \times F_{t,s}^{arch}, \tag{3}$$

where $s \in \{\text{low}, \text{central}, \text{high}\}$ indexes the screening scenario. The final per-request energy estimate is:

$$\hat{E}_{t,s} = E_a \times \left(\frac{P_{t,s}^{eff}}{P_a} \right)^{\alpha_s} \times \left(\frac{V_t}{V_{ref}} \right)^{\beta_s}. \tag{4}$$

The current implementation uses $\alpha_{low} = 0.85$, $\alpha_{central} = 0.95$, $\alpha_{high} = 1.05$, and $\beta_{low} = 0.85$, $\beta_{central} = 0.92$, $\beta_{high} = 1.0$. Carbon is then derived from the retained country mix:

$$\hat{C}_{t,s,c} = \frac{\hat{E}_{t,s}}{1000} \times CI_c, \tag{5}$$

where CI_c is the carbon intensity of country c in $\text{gCO}_2\text{e/kWh}$.

This method is not presented as a physical law of inference scaling. It is a traceable screening proxy anchored in observed prompt energy and explicit contextual assumptions. Its purpose is to provide a fast and inspectable estimate for comparative reasoning, not an audited declaration of real provider-side energy use.

3 Training Screening Method

The observatory also exposes training orders of magnitude for current market models. Here the uncertainty is even higher than for inference because published values are sparse, heterogeneous, and often reported only as aggregate emissions. A pure parameter-only scaling is therefore too brittle. The current release instead uses a second bounded proxy that combines retained parameter count with additional training priors.

The training proxy starts from literature anchors that directly report training CO_2e, and from training-energy reconstructions derived from those emissions when the source-country electricity mix is documented. For a target model t, the central training estimate is:

$$\hat{E}_{t,s}^{train} = E_a^{train} \times \left(\frac{P_t}{P_a} \right)^{\alpha_s} \times \left(\frac{Tok_t}{Tok_a} \right)^{\beta_s} \times F_{t,s}^{reg} \times F_{t,s}^{arch-tr} \times F_{t,s}^{hw}, \tag{6}$$

where P_t is the retained parameter count, Tok_t is a training-token prior, F^{reg} is a training-regime prior, $F^{arch-tr}$ captures architecture and multimodality assumptions, and F^{hw} is a hardware-class proxy. The current release uses a simple prior of 20 training tokens per retained parameter when no better public estimate is available, and defaults market models to a foundation-pretraining regime unless a narrower public indication exists.

This second proxy is not used in the application estimator shown to users, which remains inference-only. It is used in the observatory and the comparative tables to avoid suggesting that training impacts can be projected from parameter count alone. The result should still be read as a screening order of magnitude rather than as an audited declaration of model-development impact.

4 Results and Comparative Outputs

The public observatory standardizes model comparison under one hour of active use corresponding to 1,000 input tokens, 550 output tokens, one LLM request per interaction, and approximately 34.6 interactions per hour, derived from a reading-speed convention [3]. The underlying dataset and derived tables were refreshed for the March 2026 release, so the published values mirror the recalculated inference and training columns for the expanded market-model catalog. This provides a common comparison space before application-specific annualization.

Table 1 summarizes a few illustrative outputs. The first block shows how the standardized observatory makes model orders of magnitude legible. The second block shows how the same proxy can be annualized at the software-feature level from compact, natural-language scenarios.

Case	Energy	Carbon	Interpretation
Ministral 3B, one active hour	0.19 Wh	0.0077 gCO_2e	Small hybrid model under a French provider-country proxy
GPT-5 mini, one active hour	5.90 Wh	2.272 gCO_2e	Medium proprietary hosted model under a US provider-country proxy
Claude Opus 4.1, one active hour	103.53 Wh	39.860 gCO_2e	Very large proprietary estimate illustrating the steep growth of screening orders of magnitude
Support chatbot, Ministral 8B, 20,000 conversations/month	2.38 kWh/year	96 gCO_2e/year	Low-carbon annual result despite high usage because the retained electricity factor is favorable
Retrieval assistant, GPT-5 mini, 4,000 uses/month	12.31 kWh/year	4.74 $kgCO_2e$/year	Higher annualized result driven by hosted-service assumptions and US electricity contextualization

Table 1: Illustrative outputs from the current observatory and application estimator. These values are screening estimates, not audited declarations.

These examples illustrate three practical properties of the method. First, annualization matters: apparently small unit values become relevant when multiplied by real usage. Second, the chosen model profile matters materially. Third, the retained electricity mix can dominate the carbon interpretation, which is why country contextualization is explicitly shown in the interface. Just as importantly, these results can be obtained quickly from natural-language inputs while keeping the extracted scenario and assumptions visible in the interface.

For completeness, Table 2 reports the current calculated values for all 40 market models tracked in the March 2026 release. The purpose of this table is documentary: it makes the observatory outputs directly inspectable in the paper, while the web interface remains the primary medium for interactive exploration.

Table 2: Calculated environmental indicators for the 40 market models currently tracked by ImpactLLM. Inference values correspond to the standardized one-hour scenario used in the observatory. Training values are multi-factor screening estimates.

Model	Inference Wh/h	Inference gCO_2e/h	Training GWh	Training $ktCO_2e$
Claude Opus 4.1	103.53	39.86	6900.35	6166.82
GPT-5.2	96.52	37.16	5589.28	4995.13
GPT-5.2-pro	96.52	37.16	5589.28	4995.13
Grok 4	38.04	14.65	621.03	555.01
Megatron-Turing NLG 530B	26.24	10.10	795.04	710.53
GPT-4	26.03	10.02	819.76	732.62
Claude Sonnet 4	22.44	8.64	276.01	246.67
Grok 2	19.69	7.58	155.26	138.75
Llama 3.1 405B	19.13	7.37	287.06	256.54
Gopher 280B	14.31	2.57	126.01	112.61
Gemini 2.5 Pro	12.46	4.80	69.00	61.67

Model	Inference Wh/h	Inference gCO_2e/h	Training GWh	Training $ktCO_2e$
Grok 1	10.72	4.13	226.42	202.35
Claude 3.5 Sonnet	10.23	3.94	52.83	47.21
Claude 3.7 Sonnet	10.23	3.94	52.83	47.21
Jurassic-1 Jumbo	9.30	3.58	77.88	69.60
GPT-3.5 Turbo	9.16	3.53	53.60	47.90
OPT 175B	8.09	3.11	61.25	54.74
LaMDA 1	7.26	2.79	30.83	27.55
Mistral Large	7.18	0.29	26.10	23.32
GLaM 130B	6.38	2.46	35.10	31.37
Claude 2	6.01	2.31	17.25	15.42
GPT-5 mini	5.90	2.27	15.57	13.91
GPT-5 nano	5.90	2.27	15.57	13.91
Gemini 2.5 Flash	5.22	2.01	11.04	9.87
Qwen2.5 72B	3.71	2.00	9.07	8.11
Llama 3.1 70B	3.61	1.39	8.58	7.66
DeepSeek R1	2.89	1.56	675.40	603.60
DeepSeek V3	2.76	1.49	675.40	603.60
Gemini 2.5 Flash-Lite	2.06	0.79	1.55	1.39
Gemini 2.0 Flash	2.06	0.79	1.55	1.39
Qwen2.5 32B	1.72	0.93	1.79	1.60
Mistral Small 3.1 24B	1.43	0.06	1.10	0.99
Claude 3.5 Haiku	1.43	0.55	0.83	0.75
Codestral 22B	1.20	0.05	0.81	0.72
Ministral 8B	0.49	0.02	0.11	0.10
Llama 3.1 8B	0.46	0.18	0.11	0.10
Qwen2.5 7B	0.40	0.22	0.09	0.08
GPT-OSS 120B	0.40	0.16	21.56	19.27
GPT-OSS 20B	0.26	0.10	0.69	0.62
Ministral 3B	0.19	0.01	0.02	0.01

5 Literature and Source Base

The results above are not produced from a single benchmark, but from a layered source base combining direct environmental anchors, estimator literature, infrastructure and electricity context, and broader analytical framing. This structure corresponds to the source logic exposed in the site's *Sources* page, where each value is linked to a document, a system boundary, and a use role in the estimator.

5.1 Direct Environmental Anchors for LLMs

The primary literature anchors for training and inference come from work that reports environmental indicators directly for language-model systems or adjacent production systems. The core training anchors include Strubell et al. on transformer training costs [28], Luccioni et al. on BLOOM [15], Morrison et al. on holistic language-model impacts [19], and recent lifecycle or training-oriented extensions such as Fernandez et al. [9] and d'Orgeval et al. [6]. For inference, the most operational prompt-level anchor in the current release is Elsworth et al. [8], complemented by broader framing from Ren et al. [24] and water-related contextualization from Li et al. [14].

5.2 Measurement, Tracking, and Proxy Literature

The methodological design also builds on the literature and tooling ecosystem for measuring or estimating AI impacts when direct telemetry is available or partially missing. This includes CarbonTracker [2], CodeCarbon [18], the ML.ENERGY benchmark [26], and proxy-oriented tooling such as EcoLogits [10]. These references do not provide one transferable number for all current hosted LLMs, but they define the design space within which a screening proxy must remain explicit about assumptions, units, and uncertainty.

5.3 Infrastructure, Electricity, and Cloud Context

A second family of sources informs the infrastructural and contextual layers needed to convert energy proxies into carbon estimates or to interpret data-center scale effects. This includes cloud and infrastructure accounting references such as Cloud Carbon Footprint [4], AWS CCFT [1], the Microsoft Emissions Impact Dashboard [17], Google Cloud carbon reporting [11], and the OVHcloud environmental tracker [20]. It also includes system-level or electricity-demand framing from EPRI [7], Lawrence Berkeley National Laboratory [13, 27], the International Energy Agency [12], and Joule-scale discussion by de Vries-Gao [5].

5.4 Model Documentation and Conceptual Framing

Some source families play a different role: they do not provide direct environmental measurements but document model properties or frame the interpretation of the results. Examples include model and provider documentation such as the Llama 3.1 model card [16], broader environmental framing from Rillig et al. [25], and earlier conceptual work on sustainable AI and environmental decision support [22, 21]. The observatory and the site itself are part of this transparency layer because the project republishes those linked records in reusable interfaces [23].

5.5 Use-Phase and Comparison Conventions

Finally, a few references are used to define common use scenarios or contextual comparison devices rather than model-specific environmental values. In the current release, Brysbaert [3] is used to anchor the reading-speed convention behind the standardized one-hour observatory scenario. This kind of contextual reference matters because comparative results depend not only on environmental anchors, but also on the retained usage normalization and interpretation framework.

6 Discussion and Perspectives

Beyond the catalog table and the static release timelines, the observatory can also be read as a dynamic screening signal about the acceleration of frontier-model impacts. Figures 1 and 2 summarize that intuition with a simple log-linear reading of the GPT, Claude, and Grok flagship series. The purpose is not to claim a physical law or a universal growth constant. It is to provide a compact interpretation of the retained central estimates under the current observatory assumptions.

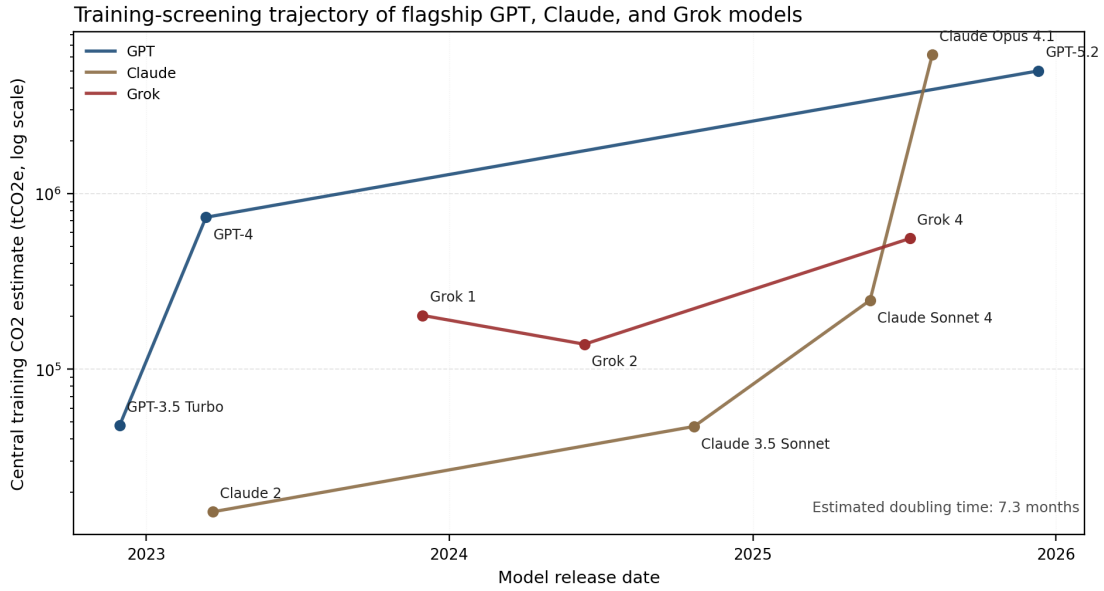

Figure 1: Illustrative reading of the observatory's central training-screening estimates for flagship GPT, Claude, and Grok models. The visual condenses the current retained values into an approximate doubling-time interpretation.

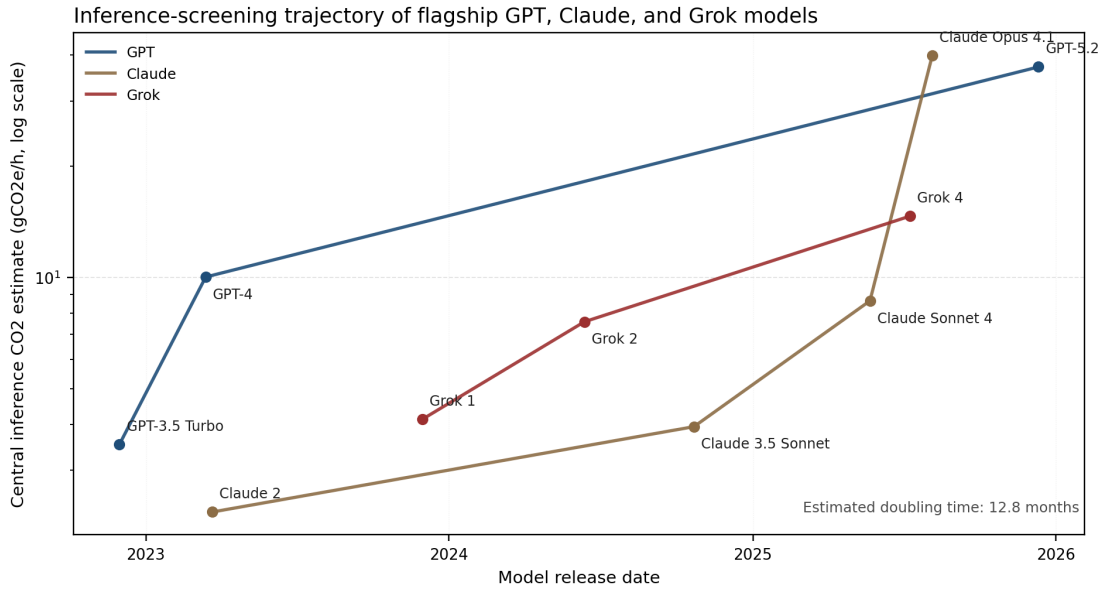

Figure 2: Illustrative reading of the observatory's central inference-screening estimates for flagship GPT, Claude, and Grok models, using the standardized one-hour scenario retained by the site.

Two lessons follow from these perspective views. First, the apparent acceleration is stronger for training than for inference. This is consistent with the current screening structure: training combines retained parameter count, token priors, architecture effects, and hardware assumptions, so frontier-model scaling quickly reaches very high orders of magnitude. Inference, by contrast, is partially damped by the standardized use scenario and by the fact that active compute per request does not necessarily grow as fast as total model size.

Second, these visuals are useful precisely because they remain interpretive rather than declarative. They help communicate that the observatory does not only contain isolated point estimates; it also suggests a market trajectory in which the environmental stakes of frontier models may be rising rapidly. But the value of that reading depends entirely on the transparency of the underlying assumptions. If those assumptions change, the apparent doubling times change as well. The figures should therefore be read as discussion tools and policy prompts, not as direct provider-side measurements or immutable forecasting laws.

From a policy and governance perspective, this is arguably one of the most useful roles an open observatory can play. Even without direct industrial telemetry, it can surface the possibility that model-scale growth is outpacing the public visibility of environmental reporting. In that sense, the discussion value of the observatory is not limited to ranking models; it also lies in making structural trends legible enough to motivate better disclosure, more granular reporting, and more disciplined debate.

7 Limitations

The main limitations are methodological. First, the estimator depends on scarce literature anchors, so the uncertainty intervals rely on wide parameter- and token-exponent bounds to keep the screening range honest. Second, proprietary service configurations remain opaque, forcing the method to rely on contextual proxies for regime, architecture, and hardware. Third, training-level emissions are reconstructed from sparse reports and assumed token priors; more published training anchors would reduce that uncertainty. Finally, the natural-language parser must interpret ambiguous descriptions, which is why every extracted assumption is surfaced to the user along with the final estimate.

8 Implementation Notes

The prototype is implemented as a Python/Flask web application that builds on the observatory dataset stored in `ImpactLLM/data/market_models.csv`. The front-end sends the user description to OpenAI for

parsing, then queries the estimator to compute the inference and training proxies. The same dataset drives the observatory tables, which remain independent of the estimator logic.

9 Conclusion

The contribution of this manuscript is both methodological and documentary: it explains how ImpactLLM transforms natural-language descriptions into bounded inference and training estimates, and it makes the resulting comparative tables and figures for the tracked models directly inspectable in one place.

That contribution should not be read as an attempt to eliminate uncertainty. For the most widely used proprietary LLM services, uncertainty is structurally unavoidable because the decisive telemetry remains inaccessible. The relevant methodological task is therefore to organize approximation rather than to pretend to escape it. This paper argues that a transparent, source-linked, reproducible screening proxy is a better basis for comparison and discussion than the current situation in which strong environmental claims often circulate without explicit assumptions, traceable derivation, or shared comparison rules.

References

[1] Amazon Web Services. What is the customer carbon footprint tool?, 2026. AWS documentation, accessed March 2026.

[2] Laurel Anthony, Benjamin Kanding, Raghavendra Selvan, Erik Christensen, Ole Andersson, et al. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. *arXiv preprint arXiv:2007.03051*, 2020.

[3] Marc Brysbaert. How many words do we read per minute? a review and meta-analysis of reading rate. *Journal of Memory and Language*, 109:104047, 2019.

[4] Cloud Carbon Footprint. Cloud carbon footprint methodology, 2026. Open-source software documentation, accessed March 2026.

[5] Alex de Vries-Gao. Artificial intelligence: Supply chain constraints and energy implications. *Joule*, 9(6):1153–1156, 2025.

[6] Theo d'Orgeval, Célia Azaïs, Jean-Louis Michaux, Guillaume Perrin, Valentin Trévisan, et al. Life cycle assessment of data centres for generative artificial intelligence. *Applied Energy*, 392:126617, 2026.

[7] Electric Power Research Institute. Power demand for data centers and artificial intelligence, 2024.

[8] Catherine Elsworth, Kuan Huang, David Patterson, Ian Schneider, et al. Measuring the environmental impact of delivering ai at google scale. *arXiv preprint arXiv:2508.15734*, 2025.

[9] Celine Fernandez, Luis Pérez-Lombard, Gonzalo Ruiz, Eduardo Gutiérrez, et al. Life cycle assessment of large language models: A methodological proposal. *Tackling Climate Change with Machine Learning*, 2025.

[10] GenAI Impact. Ecologits documentation and methodology, 2026. Open-source software documentation, accessed March 2026.

[11] Google Cloud. View carbon emissions reports, 2026. Product documentation, accessed March 2026.

[12] International Energy Agency. Energy and ai, 2025.

[13] Lawrence Berkeley National Laboratory. Report evaluates increase in electricity demand from data centers, 2025.

[14] Pengfei Li, Jianyi Yang, Md Arafat Islam, and Shaolei Ren. Making ai less "thirsty": Uncovering and addressing the secret water footprint of ai models. *Communications of the ACM*, 68(4):46–54, 2025.

[15] Alexandra Sasha Luccioni, Sebastien 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.

[16] Meta. Llama 3.1 model card, 2024.

[17] Microsoft. Emissions impact dashboard, 2026. Product documentation, accessed March 2026.

[18] MLCO2 Project. Codecarbon documentation and methodology, 2026. Open-source software documentation, accessed March 2026.

[19] Jack Morrison, Christine Na, Jose Fernandez, Tim Dettmers, et al. Holistically evaluating the environmental impact of creating language models. *arXiv preprint arXiv:2503.05804*, 2025.

[20] OVHcloud. Ovhcloud launches environmental impact tracker, 2024. Product announcement.

[21] Arnault Pachot and Céline Patissier. Toward sustainable artificial intelligence: An overview of environmental protection uses and issues. *Green and Low-Carbon Economy*, 3(2):105–112, 2023.

[22] Arnault Pachot, Céline Patissier, and Open Studio. *Intelligence artificielle et environnement : alliance ou nuisance ? L'IA face aux défis écologiques d'aujourd'hui et de demain*. Dunod, 2022.

[23] Arnault Pachot and Thierry Petit. Impactllm, 2026. GitHub repository and online demo: https://dev.emotia.com/impact-llm.

[24] Siyuan Ren, Bill Tomlinson, R. W. Black, and Andrew W. Torrance. Reconciling the contrasting narratives on the environmental impact of large language models. *Scientific Reports*, 14:28180, 2024.

[25] Matthias C. Rillig, Martina Ågerstrand, Min Bi, Kerry A. Gould, and Uli Sauerland. Risks and benefits of large language models for the environment. *Environmental Science & Technology*, 57(9):3464–3466, 2023.

[26] Nikit Sharma, Hrishikesh Chawla, Lore Mansfield, Harjot Singal, Bilge Acun, et al. The ml.energy benchmark: Toward automated inference energy measurement and optimization. *arXiv preprint arXiv:2505.06371*, 2025.

[27] Arman Shehabi, Andrew Newkirk, Sarah J. Smith, Alex Hubbard, Nhat Lei, et al. 2024 united states data center energy usage report. Technical report, Lawrence Berkeley National Laboratory, 2024.

[28] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, 2019.