# Fair, Practical, and Efficient Carbon Accounting for LLM Serving


Yueying Lisa Li*
Cornell, Cornell Tech

Leo Han
Cornell Tech

G Edward Suh
NVIDIA and Cornell

Christina Delimitrou
MIT

Fiodar Kazhamiaka
Azure Research - Systems

Esha Choukse
Azure Research - Systems

Rodrigo Fonseca
Azure Research - Systems

Liangcheng Yu
Microsoft Research

Jonathan Mace
Microsoft Research

Udit Gupta
Cornell Tech



## ABSTRACT

We propose a framework for evaluating carbon attribution methods for multi-tenant LLM serving. The framework formalizes the problem using three key components: (1) a set of requests with varying prompt and decode lengths, (2) the LLM inference runtime including batching algorithms, and (3) a carbon emission model accounting for both operational carbon (proportional to power consumption and carbon intensity) and embodied carbon from hardware manufacturing. Using the Shapley value as ground truth for fair attribution, we demonstrate why simple 'leave-one-out' attribution methods fail to satisfy efficiency properties. The framework evaluates attribution methods against four criteria: scalability (computational complexity), fairness (minimizing deviation from Shapley values), sample efficiency (algorithmic approximations for complex cases), and incentivization (encouraging users to optimize their usage patterns).


## 1. INTRODUCTION

The exponential growth of Large Language Model (LLM) applications has created unprecedented computational demands on cloud infrastructure [8, 21]. As these workloads surge, they increasingly challenge the sustainability commitments made by public cloud providers. While training large models receives significant attention, the continuous nature of inference operations represents the dominant long-term carbon contributor. This reality creates an urgent need for accurate carbon attribution mechanisms that can provide visibility into environmental impacts and support various downstream LLM-empowered applications, such as AI agents during inference time [3, 29, 28, 31].

Despite growing recognition of this need, current approaches to attributing carbon emissions in LLM inference workloads remain inadequate and often fail to account for the complex, multi-tenant nature of modern deployment environments. A naive approach to carbon attribution might allocate emissions proportionally to tokens processed in different phases (e.g., decode vs. prefill) for requests sharing the same execution backend, similar to OpenAI's pricing model [19]. Another approach is to follow traditional carbon or price attribution that uses time-based and utilization-based methods [10, 9, 5]. However, such methods break down in practical scenarios. As a simple example, decoding time is longer per scheduling iteration when the context is longer, which contributes to larger carbon footprint due to more memory bandwidth overhead for fetching larger KV cache: if we treat the 1000th token the same as the first few decoding token, it would lead to users underestimating the carbon impact of having long generation sequences, especially for complex reasoning workloads [13]. This means that a token-based strawman approach would not work well (Section 4.2).

The goal of this paper is to establish a principled framework for carbon accounting in LLM serving systems. We ask: *What properties are essential for effective carbon attribution frameworks in multi-tenant LLM environments?* Our contribution is a systematic approach that evaluates carbon attribution methods through four critical dimensions: **scalability, fairness, sample efficiency, and incentivization**. Rather than proposing a single attribution solution, we provide a comprehensive evaluation framework that enables researchers and practitioners to develop carbon accounting methods that balance accuracy, fairness, and practicality while accounting for the unique challenges of LLM inference workloads in shared environments.

## 2. MOTIVATION

### 2.1 Need for LLM Carbon Attribution


*Corresponding author. Major work done during an internship with Microsoft, further supported through NSF PPoSS and Access Grants. We are grateful for Prof. Eva Tardos and Andrea Lodi for the insightful discussion.



CarbonMetrics, ACM SIGMETRICS 2025, Stony Brook, NY, USA

Multiple recent studies emphasize that inference plays a significant role in the overall carbon lifecycle of LLMs [**?**, 15]. Notably, Jegham et al. [12] benchmarked 30 commercial LLMs and found that a single short GPT-4o query consumes approximately 0.43Wh, whereas more complex long-prompt models (e.g., o3, DeepSeek-R1) may expend over 33 Wh—over 70× more energy per request. Joanna Stern's field measurements corroborate this variability, reporting that GPT-4 text prompts range from roughly 0.17 to 1.7 Wh depending on infrastructure and model size [25]. Taken together, these findings illustrate that per-query energy usage spans two orders of magnitude—highlighting that reliable carbon accounting demands granular, workload-level measurement or more advanced techniques rather than coarse, uniform estimates.

LLM inference APIs differ widely in their architectural design. For instance, some APIs (e.g., OpenAI, Anthropic) operate fully-managed, opaque inference endpoints, while others like vLLM or Hugging Face Transformers offer more customizable open-source backends with tunable scheduling and caching policies. These differences influence hardware utilization, latency, and energy efficiency per request. Moreover, proprietary APIs may abstract away infrastructure decisions (e.g., GPU model, colocation, power capping), making it difficult to assign fine-grained carbon responsibility to end users or workloads.

The discrepancy between the actual carbon impact of a workload and its attributed emissions represents a significant challenge in cloud carbon accounting. This mismatch can lead to several issues:

- Inaccurate reporting: Organizations may over- or under-report their carbon emissions based on flawed attribution, potentially affecting their compliance with environmental goals or regulations.
- Unfair cost distribution: In scenarios where carbon emissions are tied to TCO (e.g., buying carbon offsets), inaccurate attribution could lead to unfair cost distribution among tenants.

As regulatory pressure for environmental reporting increases, providers need this framework to deliver transparent, defensible carbon attribution that aligns economic and sustainability incentives across their customer base.

## 2.2 Challenges of LLM Carbon Attribution

Our framework primarily serves cloud providers and LLM API services operating at a massive scale. Consider a provider handling billions of daily inference requests across thousands of tenants sharing the same infrastructure. In such environments, traditional attribution methods fail to capture the complex dynamics of multi-tenant LLM serving. For instance, when an e-commerce company's customer service chatbot (short prompts but high request volume) shares GPUs with a legal firm's document analysis service (long prompts with low request volume), attributing request-level carbon with low overhead becomes challenging for the following reasons:

**No ground truth execution time.** In multi-tenant environments, resource sharing and system optimizations make it challenging to determine the true execution time of individual tasks, complicating the baseline establishment for carbon attribution.

**Multi-dimensional resources sharing.** LLM inference poses new challenges compared to traditional cloud workloads or training workloads. For example, earlier works consider only a single tenant using an LLM backend [7, 23, 26]. In reality, different users' LLM API calls can be batched together, complicating the attribution across different user requests.

**Dynamic batching complexity.** Continuous batching techniques, while improving efficiency, complicate attribution as requests are dynamically grouped and processed together [30].

**Non-linear energy and latency.** The non-linear energy savings from batching challenge conventional attribution methods that assume additive resource consumption (Figure 1).

Since there are many different ways to design an LLM inference carbon attribution method, we take a principled approach by first analyzing the metrics needed to evaluate those methods.

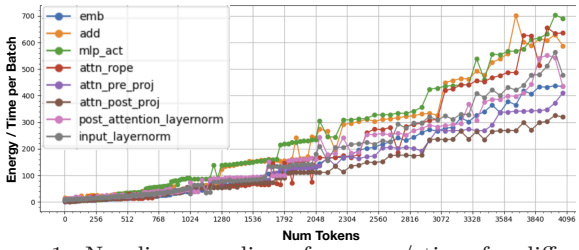


Figure 1: Non-linear scaling of energy / time for different LLM inference operators across various token numbers in a batch during request serving on Llama-8B model averaged across 10 runs on A100-SXM4-80G. It is evident that simple latency-based attribution fails.

# 3. ATTRIBUTION METRICS

## 3.1 Evaluating Fairness & Accuracy

**The Shapley value as a ground truth for fairness.** Since formulated in 1951 by Lloyd Shapley [24], **the Shapley value** has been the golden standard for solving fair attribution problems [18]. The Shapley value is unique in that it allows an intuitively fair attribution solution, as defined by adherence to the four fairness properties: **(1) Null Player**—null requests (i.e. empty requests) are attributed no carbon, **(2) Symmetry**—the same requests that are served at the same time are attributed the same amount of carbon, **(3) Efficiency**—the entire carbon footprint of the LLM inference system is fully attributed across served requests, **(4) Linearity**—attributing carbon separately for different parts (i.e nodes, node partitions) of the LLM inference system results in attributions consistent with attributing across the entire system at once.

The Shapley value has a rich history of being used as a ground truth for fair attribution across many domains such as environmental economics [20, 4], public infrastructure [16, 14, 27], networking and telecommunications [2], explainable machine learning [17, 22], and energy attribution for computer systems [6, 11].

**The Shapley distance as a measure of fairness.** With

the Shapley value as the ground truth for fair attribution, **the Shapley distance**, as defined by the deviation between attribution and the Shapley value solution, can be used as a quantitative measure of fairness [1]. In the case of carbon attribution, the Shapley distance has units of $CO_2e$.

**Fairness**: We use Shapley value as the ground truth fair carbon attribution. The relative Shapley distance (i.e. percentage deviation for the Shapley) will be used as a measure of fairness to evaluate carbon attribution methods.

## 3.2 Evaluating Practicality

In addition to Shapley-based fairness, we propose the following quantifiable properties for practical evaluation:

1. **Scalability**: We define computational scalability through *Algorithmic complexity*: The time complexity with respect to the numbers of clients ($C$), scheduling iterations ($I$), total tokens ($t$), and number of requests ($R$) should be under $\text{POLY}(C, I, t, R)$.

2. **Sample efficiency**: We evaluate algorithmic efficiency using *sample efficiency*—the number of samples $N_\epsilon$ required to achieve an error bound $\epsilon$ when approximating Shapley values: $N_\epsilon = \min\{N : \mathbb{P}(|\hat{\phi}_i - \phi_i| \leq \epsilon) \geq 1 - \alpha\}$ where $\hat{\phi}_i$ is the approximated Shapley value, $\phi_i$ is the true Shapley value, and $\alpha$ is the confidence level.

3. **Incentivization**: We quantify incentivization through a practical, directly measurable metric: *reduction reward*. For example, when the prompt length is reduced by 50%, we can quantify prompt length reduction reward as the percentage reduction of the attributed carbon:

$$R_{\text{prompt}} = \frac{C(p) - C(p/2)}{C(p)} \times 100\% \tag{1}$$

   where $C(p)$ is the carbon attributed with prompt length $p$. Higher values indicate stronger incentives to optimize prompts. The larger the reward, the more incentivization it provides for the users of the APIs. In a reasoning model like DeepSeek-R1, a more carbon-conscious user may preempt the running chain of thought during the decoding phase, and it will have a similar formulation.

# 4. CASE STUDIES

## 4.1 Problem Formulation

A natural formulation of the problem is to consider the carbon attribution of requests being served in the same LLM backend from different users/applications. There are three ingredients to the problem.

First, the Let $R = \{(p_i, d_i)\}_1^n$, where $p_i, d_i$ denotes the prompt length and decode length of request $i$ ($R_i$).

The second component is the LLM inference runtime, including the batching or scheduling algorithm together with the model itself, which we denote as system function $S$. This is a latent variable internal to the LLM serving framework. Any improvement on $S$ will potentially change the carbon attribution for a fixed request or request group.

The third component is the overall carbon emission model $C = C^{emb} + C^{op}$, where the operational carbon $C^{op} = \int_{t_0}^{t_1} P(t)CI(t)\,dt$ is proportional to the power and energy supply carbon intensity (CI), and $C^{emb} = \int_{t_0}^{t_1} eci(t)\,dt$ is proportional to the embodied carbon intensity (eci) which is a factor of hardware manufacturing carbon and the lifetime of the hardware.

## 4.2 Why Simple Attribution Fails

We present two case studies where simple carbon attribution methods can fail, particularly under fairness criteria.

**Strawman's approach 1 (LOO)** A straightforward approach to quantify each request $i$'s carbon impact is to assess its marginal contribution to the total emissions: $\phi_i = C(R) - C(R - \{i\})$. This formula represents the difference in carbon emissions when including versus excluding $R_i$. Known as the 'leave-one-out' (LOO) method, this approach calculates how much the overall carbon footprint would decrease if a specific request were omitted.

Consider a GPU batch serving system with two requests coming in at the same time with the same prompt, and $C(\emptyset) = 30,\quad C(\{r_i\}) = 40, \forall i \in \{1, 2\}$ , $C(\{r_1, r_2\}) = 42$.

Here we attribute only the GPU operational carbon with fixed carbon intensity (CI). However, this attribution technique falls short of meeting efficiency criteria. Under LOO, the carbon attributed to either $r_1$ and $r_2$ is 2. However, $30 + 2 + 2 \neq 42$ which violates the Shapley's efficiency property when serving two requests together, being unfair for the cloud provider. This makes the cloud provider taking more carbon ($42 - 4 = 38$) though the runtime system didn't change compared with a zero load scenario (30).

**Strawman's approach 2 (TBA)** Another way is to use token based attribution (TBA). To attribute the carbon in proportion to the request's weighted token length $\phi_i = W(p_i, d_i)C(R)$, where $W(p_i, d_i) = w_1 p_i + w_2 d_i,\ \sum_{i=1\ldots n} W(p_i, d_i)C(R) = C(R)$.

Consider a GPU batch serving system with two requests $r_1, r_2$ arriving at the same time with prompts of the same length, but $r_2$ generates more decoding tokens. $C$ denotes the carbon for decoding phase. $C^a$ denotes the carbon per unit time for decoding phase when the two requests are batched together until one finishes, and $C^b$ denotes the carbon per unit time after either one of the requests finish before the GPU becomes idle again. Then $C = T_a C^a + T_b C^b$. The idle carbon per unit time is $C(\emptyset) = 30,\quad C^a(\{r_1, r_2\}) = 44,\ C^b(\{r_2\}) = 40$. Assume $T_b = 2T_a,\ d_2 = 3d_1$.

We examine if the method satisfies the linearity property in Sec 3. Assume it satisfies the linearity property. By definition, it has proportionality to decoding tokens during the batched phase $C^a$, where $W = \frac{(0,1)}{d_1 + d_2}$.

Using TBA, we can calculate

$$\begin{aligned}\phi_1(C) &= \phi_1(T_a C^a + T_b C^b) && \text{(Definition of C)}\\ &= T_a\phi_1(C^a) + T_b\phi_1(C^b) && \text{(Linearity)}\\ &= T_a\phi_1(C^a)\\ &= \frac{T_a}{2}(C^a(\{r_1, r_2\}) - C^a(\emptyset)) = 7T_a && \text{(Proportionality)}\end{aligned}$$

However, if we consider

$$\begin{aligned}\phi_1(C) &= \frac{d_1}{d_1+d_2}(T_a C^a + T_b C^b)(\{r_1, r_2\} - \emptyset) && \text{(Proportionality)}\\ &= \frac{d_1}{4d_1}(T_a \times 44 + 2T_a \times 40 - 3T_a \times 30)\\ &= \frac{34T_a}{4} = 8.5T_a && \text{(Linearity)}\end{aligned}$$

The above contradicts each other; hence, the TBA doesn't satisfy linearity. □

**Notes on the other criteria:**

- **Fairness**: *LOO* violates efficiency and fairness by misallocating idle carbon to the provider, whereas *TBA* fails linearity and proportional fairness, especially in varied token-length scenarios.
- **Sample Efficiency**: Both methods require relatively few computations, thus are highly efficient in a sample efficiency perspective.
- **Scalability**: *LOO* faces computational overhead as the number of requests grows (need to run the counterfactual carbon for each request), while *TBA* scales better due to straightforward proportional calculations ($O(R)$).
- **Incentivization**: Both methods offer limited incentivization for optimal behavior; they neither strongly penalize nor reward strategic token usage.

# 5. CONCLUSION

We present *LLMCarbAccountant*, a framework for evaluating carbon attribution methods for LLM serving. By highlighting the limitations of naive heuristics and leveraging cooperative game theory, our approach argues for a more principled, transparent carbon accounting metric design. We hope this work encourages carbon-aware design to become a first-class concern in future agentic AI systems.

# 6. ADDITIONAL AUTHORS

# 7. REFERENCES


[1] Algaba, E., Fragnelli, V., and Sánchez-Soriano, J. *Handbook of the Shapley Value*, 1 ed. CRC Press, Sept. 2021.

[2] Anshelevich, E., Dasgupta, A., Kleinberg, J., Tardos, E., Wexler, T., and Roughgarden, T. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing 38*, 4 (2008), 1602–1623.

[3] Anthropic. Building effective agents, 2025.

[4] Ciardiello, F., Genovese, A., and Simpson, A. A unified cooperative model for environmental costs in supply chains: the Shapley value for the linear case. *Annals of Operations Research 290*, 1-2 (July 2020), 421–437.

[5] Dong, M., Lan, T., and Zhong, L. Rethink energy accounting with cooperative game theory. In *Proceedings of the 20th annual international conference on Mobile computing and networking* (2014), pp. 531–542.

[6] Dong, M., Lan, T., and Zhong, L. Rethink energy accounting with cooperative game theory. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking* (New York, NY, USA, 2014), MobiCom '14, Association for Computing Machinery, p. 531–542.

[7] Faiz, A., Kaneda, S., Wang, R., Osi, R. C., Sharma, P., Chen, F., and Jiang, L. LLMCarbon: Modeling the end-to-end carbon footprint of large language models. In *The Twelfth International Conference on Learning Representations* (2024).

[8] Fang, J., Yu, Y., Zhao, C., and Zhou, J. Turbotransformers: an efficient gpu serving system for transformer models. In *Proceedings of the 26th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (2021), pp. 389–402.

[9] Feigenbaum, J., Papadimitriou, C., and Shenker, S. Sharing the cost of muliticast transmissions (preliminary version). In *Proceedings of the thirty-second annual ACM symposium on Theory of computing* (2000), pp. 218–227.

[10] Han, L., Kakadia, J., Lee, B. C., and Gupta, U. Towards game-theoretic approaches to attributing carbon in cloud data centers. In *Proceedings of the 2024 HotCarbon Workshop* (2024), ACM.

[11] Islam, M. A., and Ren, S. A new perspective on energy accounting in Multi-Tenant data centers. In *USENIX Workshop on Cool Topics on Sustainable Data Centers (CoolDC 16)* (Santa Clara, CA, Mar. 2016), USENIX Association.

[12] Jegham, N., Abdelatti, M., Elmoubarki, L., and Hendawi, A. How hungry is ai? benchmarking energy, water, and carbon footprint of llm inference. *arXiv preprint arXiv:2505.09598* (2025).

[13] Jin, Y., Wei, G.-Y., and Brooks, D. The energy cost of reasoning: Analyzing energy usage in llms with test-time compute, 2025.

[14] Junqueira, M., da Costa, L. C., Barroso, L. A., Oliveira, G. C., Thome, L. M., and Pereira, M. V. An aumann-shapley approach to allocate transmission service cost among network users in electricity markets. *IEEE Transactions on Power Systems 22*, 4 (2007), 1532–1546.

[15] Li, Y. L., Graif, O., and Gupta, U. Towards carbon-efficient llm life cycle. In *Proceedings of the 3rd Workshop on Sustainable Computer Systems* (2024).

[16] Littlechild, S. C., and Thompson, G. F. Aircraft landing fees: A game theory approach. *The Bell Journal of Economics 8*, 1 (1977), 186–204.

[17] Lundberg, S. M., and Lee, S.-I. A unified approach to interpreting model predictions. In *Advances in*

*Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774.

[18] Moulin, H. *Fair Division and Collective Welfare*. The MIT Press, 01 2003.

[19] OpenAI. Openai api reference. https://platform.openai.com/docs/api-reference, 2025. Accessed: 2023-11.

[20] Petrosjan, L., and Zaccour, G. Time-consistent shapley value allocation of pollution cost reduction. *Journal of Economic Dynamics and Control 27*, 3 (2003), 381–398.

[21] Pope, R., Douglas, S., Chowdhery, A., Devlin, J., Bradbury, J., Heek, J., Xiao, K., Agrawal, S., and Dean, J. Efficiently scaling transformer inference. *Proceedings of Machine Learning and Systems 5* (2023).

[22] Rozemberczki, B., Watson, L., Bayer, P., Yang, H.-T., Kiss, O., Nilsson, S., and Sarkar, R. The shapley value in machine learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22* (7 2022), L. D. Raedt, Ed., International Joint Conferences on Artificial Intelligence Organization, pp. 5572–5579. Survey Track.

[23] Schmidt, V., Goyal, K., Joshi, A., Feld, B., Conell, L., Laskaris, N., Blank, D., Wilson, J., Friedler, S., and Luccioni, S. CodeCarbon: Estimate and track carbon emissions from machine learning computing, 2021.

[24] Shapley, L. S. *Notes on the N-Person Game mdash; II: The Value of an N-Person Game*. RAND Corporation, Santa Monica, CA, 1951.

[25] Stern, J. How much energy does your ai prompt use? i went to a data center to find out. *Wall Street Journal* (2025). Measured GPT-4 text prompts at 0.17–1.7Wh.

[26] Thoughtworks. Cloud carbon footprint, 2023.

[27] Voswinkel, S., Höckner, J., Khalid, A., and Weber, C. Sharing congestion management costs among system operators using the shapley value. *Applied Energy 317* (2022), 119039.

[28] Wang, J., Wang, J., Athiwaratkun, B., Zhang, C., and Zou, J. Mixture-of-agents enhances large language model capabilities. *arXiv preprint arXiv:2406.04692* (2024).

[29] Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., Awadallah, A. H., White, R. W., Burger, D., and Wang, C. Autogen: Enabling next-gen llm applications via multi-agent conversation, 2023.

[30] Yu, G.-I., Jeong, J. S., Kim, G.-W., Kim, S., and Chun, B.-G. Orca: A distributed serving system for transformer-based generative models. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)* (2022), pp. 521–538.

[31] Zaharia, M., Khattab, O., Chen, L., Davis, J. Q., Miller, H., Potts, C., Zou, J., Carbin, M., Frankle, J., Rao, N., and Ghodsi, A. The shift from models to compound ai systems. https://bair.berkeley.edu/blog/2024/02/18/compound-ai-systems/, 2024.