Sustainable LLM Benchmarking: Efficient Evaluation Optimizing Costs and Resources

Nathanaël Lemonnier[1,*], Jacques Kluska[2] and Vincent Morel[3]

[1]Strategy & Innovation, AI Hub, Schneider Electric, 160 av. des Martyrs, Grenoble, 38000, France
[2]Responsible AI, AI Hub, Schneider Electric, 160 av. des Martyrs, Grenoble, 38000, France
[3]AI Technology, AI Hub, Schneider Electric, 35 rue Joseph Monier, Rueil-Malmaison, 92800, France

Abstract

The surge and continuous delivery of novel Language Models (LMs) increase the need to compare their performance for appropriate selection for industrial applications. Leaderboard platforms rely on benchmark methodology that provides scores over the evaluation of completion of several queries, regrouped under a specific domain. However, the number of queries in these datasets is often huge, mobilising time, cost, and resources.

We developed a methodology to drastically reduce the volumetry of such dataset, with negligible loss of the baseline score. The direct implications are savings in costs, duration, and carbon emission, incorporated altogether for a sustainable and responsible approach in assessing LMs performance.

We used datasets from the HuggingFace OpenLLM Leaderboard v1 and tracked the performance score obtained from inferences with two models (gpt-3.5-turbo-1106 and gpt-4o-mini-2024-07-18). We used a local slope reduction method and the kneedle algorithm on lowess smoothing to determine the sufficient volume of queries, and associated loss in performance score. While most of the public benchmarks focus on only one metric, we incorporated four of them simultaneously (performance, cost, latency, carbon footprint). This allowed us to demonstrate that using only 40.7% of the datasets leads to saving 26 hours of inference computation and 1.818 kg CO2eq (>59% reduction), for a 0.75% relative loss on performance score precision.

Limiting the number of queries to benchmark LMs is a powerful trigger to adapt LMs consumption to a responsible and frugal use of AI and have more control over economical and ecological constraints. Our findings pave the way to consciousness about the unnecessary utilization of large volumes of queries in LMs Benchmark, and call for a mindset shift that should be applied when developing and evaluating industrial AI applications.

Keywords

Benchmark, Responsible AI, LLMs, Sustainability

1. Introduction

A vast number of Language Models (LMs) have been released from several providers[1] since the recent surge of AI, and recent advances allow novel technologies to be available, like Large Reasoning Models (LRMs), Agentic AI, etc. Matching a specific use case to the most appropriate LM can be cumbersome. Between Small LMs (SLMs), Large LMs (LLMs), LRMs, and so on, the challenge for users will be to select the best compromise between performance, latency, and cost. Moreover, generative AI models can have a significant energy and environmental footprint[2]. It is crucial to monitor and limit their environmental impact[3]. To facilitate users' decision-making, a myriad of benchmarks is available[4, 5, 6, 7, 8, 9, 10], providing a ranking of the LMs (i.e. leaderboard) based on scores computed from the evaluation of multiple completions obtained from sending queries of distinct datasets (i.e. benchmarks) to LM endpoints. The benchmarks are organized by specific domains: knowledge and language understanding, reasoning capabilities, grounding and abstractive summarization, content moderation and narrative control, coding capabilities, opinion and sentiment neutrality. Gathering the results from several benchmarks belonging to the same domain leads to an overall comprehension of LMs performance in such domain.

2nd Workshop on Green-Aware Artificial Intelligence, 28th European Conference on Artificial Intelligence (ECAI 2025), October 25–30, 2025, Bologna, Italy

*Corresponding author.

nathanael.lemonnier@se.com (N. Lemonnier); jacques.kluska@se.com (J. Kluska); vincent.morel@se.com (V. Morel)

However, ranking provided on standard leaderboards with public benchmarks may not be of interest to all users. Benchmarks are run on third-party infrastructure, with static parameters, which may not be those required by the business solutions. Latency may differ depending on the infrastructure, performance may be impacted by the API call parameters, costs may vary depending on the company pricing plans, etc. Companies will thus want to assess their own benchmarks on their own data and infrastructure to fully match their use case needs and implementations. More importantly, the volumetry of benchmarks can reach tens of thousands of queries or more, meaning that their implementation and processing are time-consuming and lead to unnecessary resource consumption. Efforts are required to reduce time-to-value and fit leaderboard to a responsible process.

Efforts have been developed in the community to reduce benchmarks, mostly relying on filtering the datasets and selecting a subset, based either on task or queries similarities[11, 12]. They also involve stratified random sampling on subsets scenario[13] or clustering[14], considering that some components of the benchmark may be less weighted. Computation savings are also considered[13].

In this paper, we developed a method that considers the whole set of queries equally, agnostically suggesting a cutoff based on multiple randomized distributions of the queries without filtering or weights. We achieved similar performance values to the original benchmark. More importantly, we did not only focus on performance score, but also considered savings on carbon emissions, cost and duration, which are crucial considerations when developing AI applications. We experimentally tested our methodology using commonly used datasets, with a common metric (accuracy) and considering single-domain benchmarking, to ease understanding (Arc-C[15], GSM8k[16], Hellaswag[17], MMLU[18], Winogrande[19], OpinionQA[20]). We used two popular models (gpt-3.5-turbo[21] and gpt-4o-mini[22]), which were selected for their relatively low carbon emission.

2. Experimental Setup

2.1. Workflow

The workflow, presented in Figure 1, bootstraps over 1,000 iterations. Each iteration uses a randomized distribution of the queries in the dataset to detect a cutoff, based on lowess smoothing of the absolute values of successive slopes computed from windows including a small number of queries. Cutoffs are then averaged, providing their values follows normal distribution. The novelty of this framework is that the subset filtered by the final cutoff is agnostic regarding the queries order, interdependency or importance.

From a given dataset and a given LM, we evaluated the completions to compute the performance score, used as a baseline, as given by accuracy (ratio of True over total). The next token randomness and probability parameters were alike for all datasets: temperature=0.7, top_p=0.95. The evaluation returned a boolean upon correctness of the completion regarding reference answer, using regex to postprocess the completion. Moreover, verbosity of input and output, cost[23], and latency were collected. Eventually, carbon emissions equivalences were computed using the Ecologits Calculator[24] Expert Mode, giving 0.00092 gCO2eq mper output token for gpt-3.5-turbo, and 0.00054 gCO2eq for gpt-4o-mini, based on location in France.

2.2. Bootstrap of Performance Score Tracking

We tracked the evolution of the performance score by computing the accuracy obtained at any successive index of the datasets. The tracking curve eventually reached a plateau, ending at the performance score, used for baseline.

Kneedle algorithm. The purpose was to identify the root of the plateau, from which the performance stops evolving. Due to the signal-like pattern of the tracking curve, making it impractical to detect the plateau, we first transformed it to allow smoothing. We fractioned the tracking curve into multiple frames of length n and computed the resulting slopes. We applied a lowess approximation on the

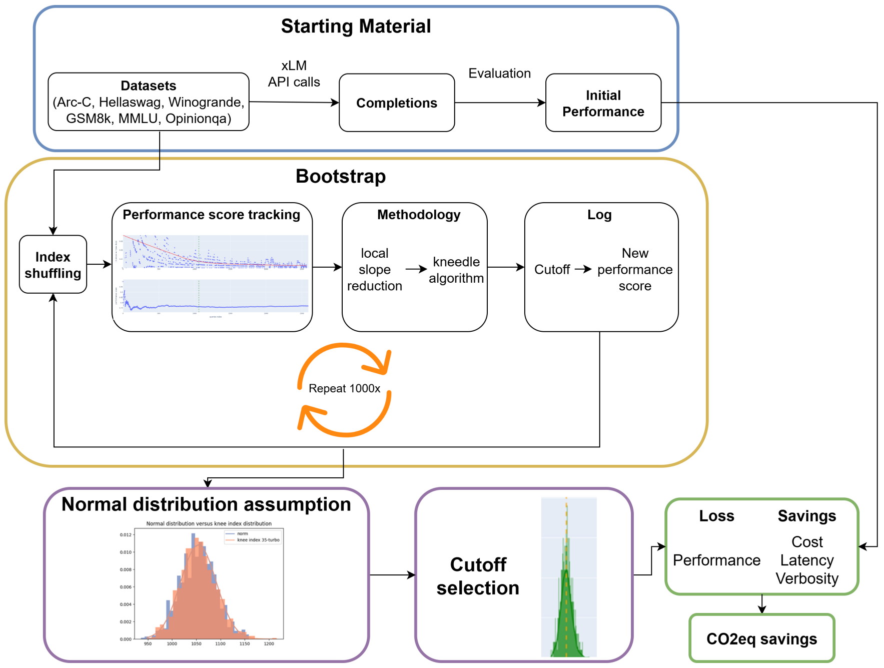


Figure 1: Worflow. A bootstrap of 1,000 iterations is used to shuffle index of datasets and compute the optimal number of queries (cutoff) and performance score according to the kneedle algorithm. The selected cutoff eventually is the mean of the 1,000 iterations cutoffs distribution, after assumption of normality. Loss on performance and savings on cost, latency, verbosity and carbon footprint are computed from the selected cutoff.

slopes values and used the kneedle algorithm[25] with appropriate curve pattern (e.g. 'concave' or 'convex') to find the knee (or elbow, respectively) as an optimal cutoff. The delta between the associated performance at cutoff and baseline performance score represents the performance loss.

Bootstrap. To ensure a negligible effect of queries order, we first shuffled the index, then tracked the performance score, logged both cutoff and performance loss, and repeated the process 1,000 times (see Figure 1).

Selected cutoff. Eventually, we verified the assumption of normal distribution of the cutoffs collected over the 1,000 iterations with a two-sample Kolmogorov-Smirnov (KS) test against a normal distribution. The selected cutoff was eventually defined as the mean of the distribution. Again, the associated performance score at the selected cutoff was compared to the baseline performance score to compute loss.

Methodology validity. Since the method requires a frame of length n to compute slopes, we evaluated the impact of n on the performance score induced by the selected cutoff (Figure 2). We tested n both as a static number, alike for all datasets, and as a relative proportion of the number of queries in the dataset (dataset length). This allowed to select an optimal frame length to compute slopes, which would a) allow to have enough points to compute slopes, b) impact a minimal loss on performance score, and c) maximise the number of slopes.

2.3. Cutoff Evaluation

We assessed robustness of the method by self-applying the cutoff to the datasets, iteratively. This allowed to understand the limitations of the dataset volumetry on which our method can be used. From the initial volumetry, we detected the cutoffs of the 1,000 bootstrap iterations, the associated deviation to normal distribution with KS p-value, and selected cutoff (mean of distribution). We cropped the dataset to the selected cutoff and re-iterated the methodology, and so on until we could reject that the distribution of 1,000 cutoffs and normal distribution were sampled from the same distribution (p-value<5e-02).

3. Results

We applied the methodology using gpt-3.5-turbo (1106) and gpt-4o-mini (2024-07-18) with adapted system prompts (See Table A1) to generate completions on the following datasets: *Arc-C* (2,590 grade-school level, multiple-choice science questions from the Challenge Set) ; *GSM8k* (8,792 grade school math problem) ; *Hellaswag* (49,947 common sense inference queries) ; *MMLU* (15,858 queries over 57 tasks like elementary mathematics, US history, computer science, law, etc. from different levels like college, high school…) ; *Winogrande* (40,490 common sense reasoning questions) ; *OpinionQA* (6,024 multiple choices queries to challenge the opinion of the LM of social subjects). The baseline performance scores are in Table 1.

3.1. Frame Length

The first step is to select a suitable frame length to compute the slopes, the lowess curve, and induce the cutoff. Using a static number of queries in the frame does not allow us to conclude (Figure 2.a.). Although the cutoff occurs around 40% of each dataset for small frames (up to 100), it drastically drops (down to 1% with 2,500-long frame for Arc-C) and is dependent on the number of queries (dataset length). Indeed, while using 2,500 queries for Arc-C leads to compute the lowess curve on only 2 slopes values, it still allows 20 slopes with Hellaswag, and reaches a cutoff at 39.5% of the dataset. A similar observation impacts the loss on performance score. While stable up to 100 queries in the frame (loss from 0.002 on Hellaswag up to 0.008 on Arc-C), the loss rises up to 0.104 (variation over 0.769, hence a 13.5% error margin) with 2,500 queries on Arc-C.

We used a frame length relative to the dataset length (Figure 2.b.) to have comparable outcomes across datasets. We observed a drop for frames larger than 5% of the dataset length. The performance loss is comparable between 1% and 5% (Arc-C: 0.008 with 1% to 0.01 with 5%, Hellaswag: 0.028 with 1% to 0.002 with 5%), as well as the percentage of the dataset concerned by the cutoff (Arc-C: 40.8% with 1% to 40.1% with 5%, Hellaswag: 40.7% with 1% to 39.8% with 5%).

We thus selected 1% of the dataset length as a satisfying length for the frames.

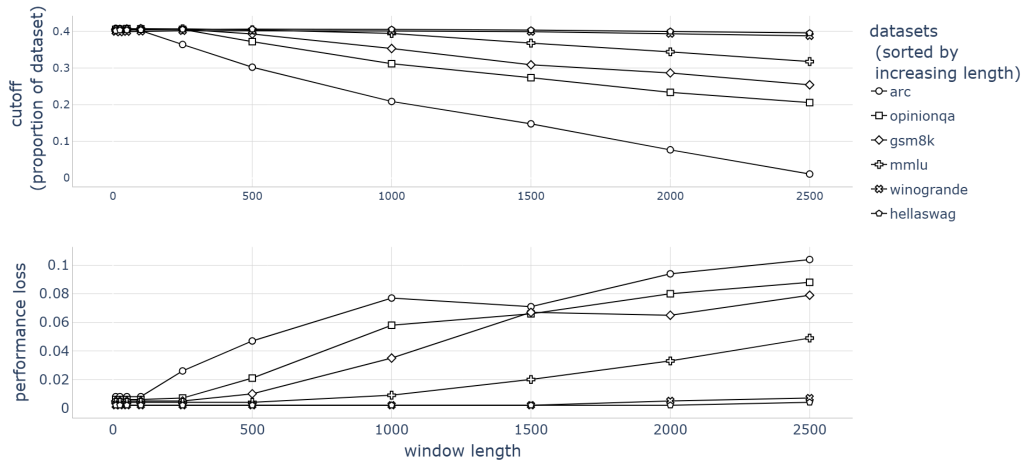
3.2. Application

An example of the methodology outcomes is presented in Figure 3: the tracking of the performance score along the queries (bottom), the tracking of the slopes (top, in blue, with n=1%), the lowess curve (top, in red) and the cutoff (dashed green line). Collecting cutoffs for all 1,000 iterations allows to build the distribution (Figure A1), which we compared to normal distribution (example on Figure 4 with Arc-C). The KS tests conclusions were in favor of similarity between normal and knee index distributions (gpt-3.5-turbo on arc-c: p=0.86 ; gpt-4o-mini: p=0.46, see Table A2). Hence, the mean of the distribution can be used as a cutoff for each dataset.

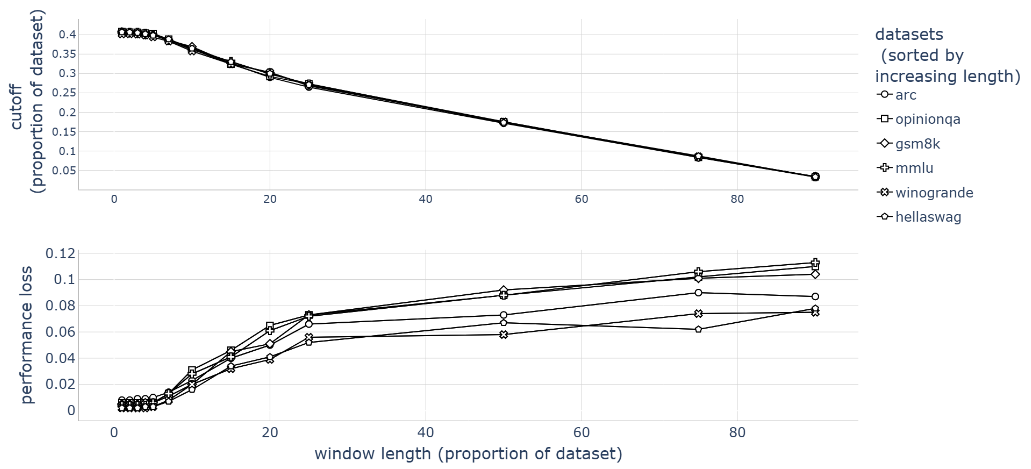
The baseline performance score, cutoff, loss on performance score and volumetry induced by the cutoff are depicted for each dataset in Table 1 (see Figure 5 for deviation from initial performance score). The method was reproducible across datasets with very consistent cutoff of 40.7% of the dataset volumetry regardless of model and initial performance score. The performance loss was also consistent with a mean relative loss of 0.75 ± 0.53% (absolute loss average is 0.43 ± 0.23% for gpt-3.5-turbo, 0.38 ± 0.18% for gpt-4o-mini).

3.3. Ecological and Economical Impact

Given the current prices for inference, the compromise between costs reduction and potential loss on performance is undisputable (Table 2). Moreover, on top of the 26 h saved to process the benchmarks for both models (8 h 42 min instead of 21 h 24 min for gpt-3.5-turbo, 8 h 55 min instead of 22 h 18 min for gpt-4o-mini), the amount of carbon emissions saved is consequent: 1.818 kgCO2eq (770.9 g instead of 1898.1 g for gpt-3.5-turbo, 464.7 g instead of 1155.3 g for gpt-4o-mini), reaching over 59% reduction.



(a) Impact of frame length on the performance loss (as a fixed number of items in the frame for all datasets)



(b) Impact of frame length on the performance loss (as a number of items in the frame dependant of each dataset length)

Figure 2: Evaluation of the impact of frame length used to compute local slopes as a fixed number of queries (A., 10, 25, 50, 100, 250, 500, 1000, 1500, 2000, 2500 queries) or as a proportion of the dataset length (B., 1%, 2%, 3%, 4%, 5%, 7%, 10%, 15%, 20%, 25%, 50%, 75%, 90%), on the cutoff to apply from the optimal index (top) as a proportion of the dataset, and on the loss of the induced performance score (bottom).

3.4. Limitations

Applying the cutoff on the same dataset iteratively ends up with very consistent results. The cutoff occurs around 40% at each iteration, regardless of the dataset size (Figure 6). However, the KS p-value reaches significance for datasets smaller than 101 queries for gpt-3.5-turbo (e.g. GSM8k, p-value=4.9e-02) or 233 for gpt-4o-mini (e.g. hellaswag, p-value=3.9e-03). Considering the iteration before reaching significance, we can observe that the relative loss in performance ranges from 3.2 to 7.4% for gpt-3.5-turbo, and from 1.1 to 11.0% for gpt-4o-mini. These observations help in understanding how many queries are required to provide significant performance score.

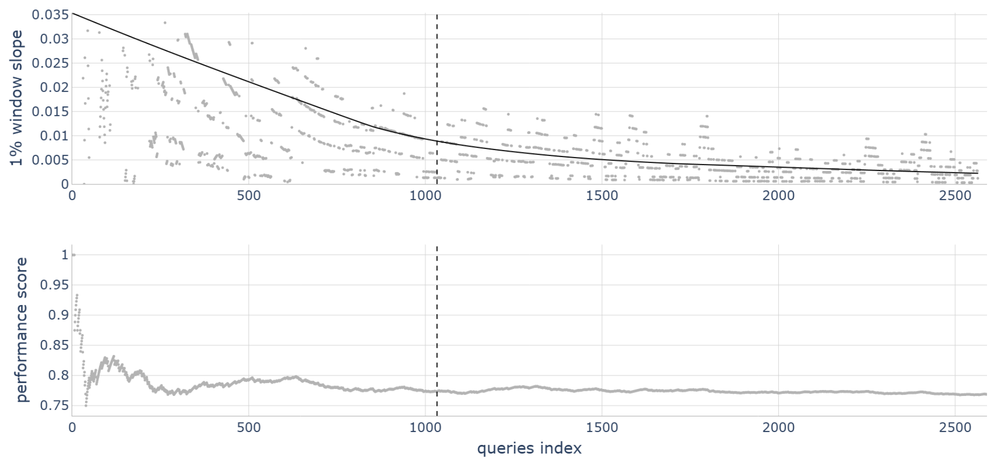


Figure 3: The tracking of the slopes (top, light grey dots) for one iteration of randomizing the Arc-C queries, computed over each successive 25-queries frames (1% of 2,590 queries), with lowess smoothing trendline (black). Tracking of the performance score (bottom) along the same queries, reaching the final performance score (.77) at last. The kneedle algorithm identifies the plateau root (dashed black line) at the 1,033th query.

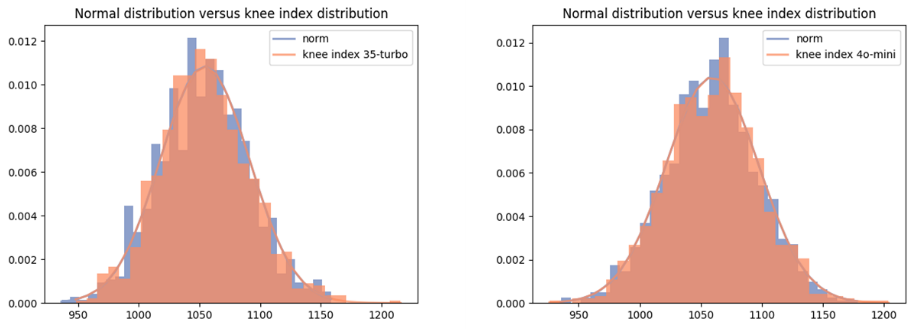


Figure 4: Distribution of the cutoffs across 1,000 iterations (light red) versus normal distribution (light blue) for gpt-3.5-turbo (left) and gpt-4o-mini (right) for Arc-C dataset.

4. Discussion

4.1. Opportunities for savings

To drastically crop the datasets for similar performance result, we used a reduction methodology on the tracking of the performance scores, for several datasets with completions generated using two popular language models. The adoption of a sustainable and responsible approach, assess carbon emission, cost and latency along performance, is a key contribution of our study. Moreover, our approach considers all queries in the datasets without subset filtering, with each query having equal weight. Additional feature could be that, for multi-domain datasets, a cutoff would need to follow constrained shuffling to respect domains proportion in the original dataset. The methodology would benefit from reproducing the results on additional datasets. However, the consistent results presented here proved that the method does not depend on the dataset. We also adapted the methodology to the TruthfulQA[26] dataset (see Figure A2), which relies on an NLP metric (e.g. BLEURT[27]) to evaluate completion over 817 open questions. The method required a normalization of the metric to adapt the BLEURT score range (from -1.81 to 1.13 range to 0-100). We obtained a cutoff at 40.7% of the dataset (333 queries) for a 1.06% loss (0.654 point difference from the final score of 61.84). Moreover, we used models that are less costly, with a certain pattern of carbon emission (0.9 and 0.5 mg per token). Confronting the latest LRM to these side-by-side would be interesting. For example, gpt-4o-mini-2024-07-18 is only 1% the

Table 1
Results of the kneedle algorithm method.

	dataset	Arc-C	GSM8k	Hellaswag	MMLU	Winogrande	OpinionQA
	queries	2590	8792	49947	15858	40490	6024
gpt-3.5-turbo	score (%)	76.9	39.7	78.7	60.8	75.8	35.5
	cutoff±std	1054±37	3582±128	20308±690	6458±226	16507±594	2455±87
	absolute loss (%)	0.78	0.50	0.18	0.37	0.21	0.59
	relative loss (%)	1.01	1.26	0.23	0.61	0.28	1.66
	% of queries	40.7%	40.7%	40.6%	40.7%	40.7%	40.7%
gpt-4o-mini	score (%)	91.7	32.1	79.1	74.0	85.1	57.7
	cutoff±std	1056±45	3576±124	20299±729	6449±224	16499±587	2453±85
	absolute loss (%)	0.50	0.49	0.18	0.33	0.16	0.61
	relative loss (%)	0.54	1.53	0.23	0.44	0.19	1.06
	% of queries	40.8%	40.7%	40.6%	40.7%	40.7%	40.7%

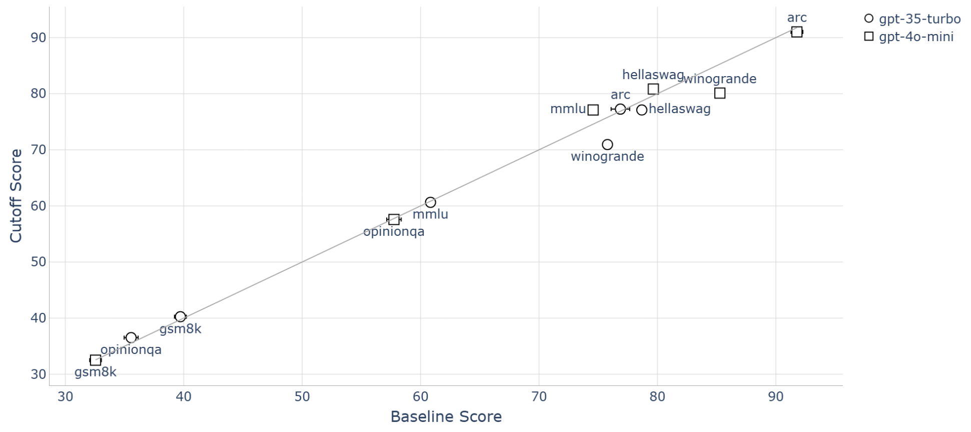


Figure 5: Deviation of the score detected with cutoff from baseline performance score for each dataset with gpt-3.5-turbo (circle) and gpt-4o-mini (square), with corresponding standard deviation.

cost of o1-2024-12-17, and 6.1% of o1 emissions per token (8.85 mgCO2eq per token in France). Using non-public datasets (data not shown), we estimated that gpt-4o-mini verbosity was 12% of o1 in low reasoning mode (when o1 uses 100 tokens to reason and generate a completion, gpt-4o-mini uses 12 tokens). Given that, in the frame of the current study, gpt-4o-mini would generate 2,135,513 tokens, the o1 equivalence would be 17,795,941 tokens, reaching a value of 157.5 kgCO2eq, and total cost of USD 318.5 for input and USD 1174.5 for output. Using our methodology would save up to 93.4kg CO2eq and USD 885.3 benchmarking o1 on these datasets. Scaling the mindset of this paper up to the whole landscape of models and leaderboard providers (possibly hundreds of models on several platforms[28]), one could leverage savings on tons of CO2eq savings and thousands of dollars.

4.2. Datasets size limitations

We studied large datasets for which volumetry reduction makes sense. However, we tested the limits of the methodology by self-applying the cutoff strategy iteratively to the same dataset. As expected, at some point, a small dataset is irrelevant since the performance loss is consequent, and the distribution of the detected cutoff for the 1,000 iterations deviates from normal distribution. Even before reaching significant non-normal distribution, the performance loss reaches over 5% for some datasets (GSM8k, Opinionqa with gpt-3.5-turbo). While the trade-off between performance loss, resource savings, and methodology robustness (normal approximation) depends on the AI task and its applications, our results roughly indicate that a dataset volume of about 1,000 queries is a minimal amount to find an

Table 2
Responsible metrics using cutoff. Initial cost, latency and GHG without cutoff are in parentheses.

	dataset	Arc-C	GSM8k	Hellaswag	MMLU	Winogrande	OpinionQA
	input tokens	273361	875364	11553269	2435920	3268834	895196
gpt-3.5-turbo	output tokens	22284	28343	1587928	167960	222821	33762
	cost (USD)	0.13 (0.32)	0.38 (0.93)	5.97 (14.7)	1.13 (2.77)	1.51 (3.71)	0.39 (0.96)
	latency (s)	510 (1254)	1842 (4526)	15835 (39003)	3315 (8146)	8586 (21096)	1221 (3001)
	GHG (gCO2eq)	8.3 (20.5)	10.6 (26.1)	593.1 (1460.9)	62.9 (154.5)	83.4 (205.0)	12.6 (31.1)
gpt-4o-mini	output tokens	23404	36992	1448205	217366	371779	37767
	cost (USD)	0.03 (0.07)	0.07 (0.18)	1.19 (2.93)	0.23 (0.57)	0.34 (0.86)	0.07 (0.18)
	latency (s)	635 (1633)	2262 (5671)	13610 (33523)	4614 (11536)	9487 (24264)	1448 (3640)
	GHG (gCO2eq)	4.9 (12.7)	8.0 (20.0)	318.1 (783.5)	47.0 (117.6)	78.6 (201.1)	8.1 (20.4)

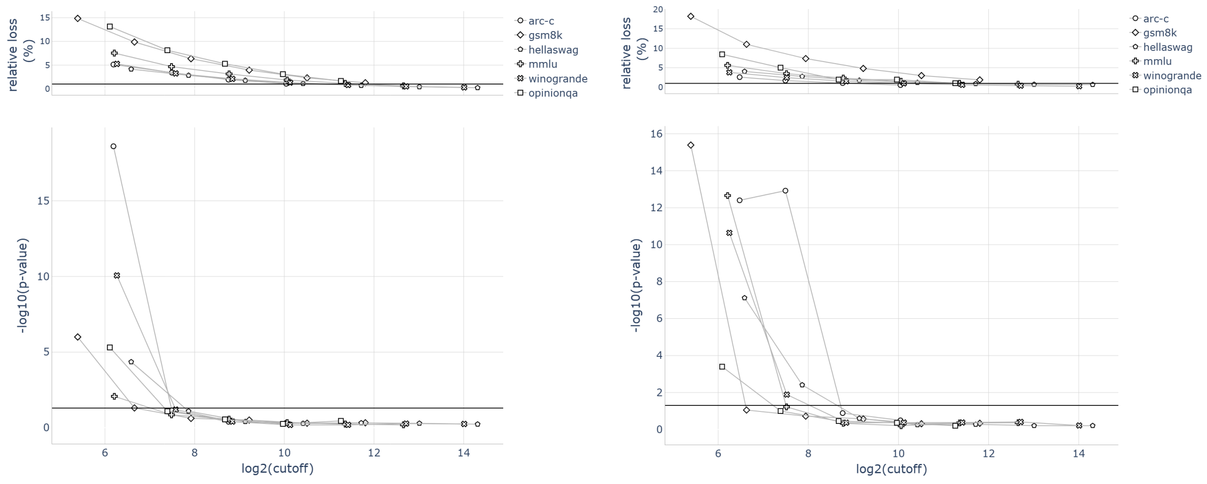


Figure 6: Impact of iterative dataset reduction for gpt-3.5-turbo (left) and gpt-4o-mini (right). Bottom: For each dataset, the cutoff was applied (x-axis, in log2) and significance of deviation from normal distribution was checked (y-axis) until significance was reached (black horizontal line, p-value = 0.05). Top: the corresponding loss relative to the baseline is indicated. The black horizontal line corresponds to 1% loss threshold.)

excellent robustness with acceptable performance loss of 1-3%. For the larger dataset (hellaswag: 49,947), respectively with gpt-3.5-turbo and gpt-4o-mini, we could iteratively reach 1,368 and 1,371 queries by keeping 1.1% and 1.2% loss on initial performance, thus using only 2.7% of the dataset. Instead of reaching 78.7 and 79.1% as initially obtained with 49,947 queries, we could achieve 77.8–79.6% and 78.1–80.0%.

4.3. Importance of parameters optimization

Although such delta between scores may not make the difference in a leaderboard for models with similar performance, one must remind that tuning parameters and system prompt remain among the important factors to enhance the performance – without mentioning in-context learning or fine-tuning techniques. While models may be tie on leaderboards score, tuning parameters, prompt, or injected knowledge will eventually have a greater impact on their performance. We chose a static set of parameters for the sake of comparison, and simple system prompts to minimize complexity ; the performance values presented here should therefore not be considered definitive. AI developers must adapt tuning to their cases.

5. Conclusion

We developed and tested a robust method to assess volume reduction for popular datasets used in public benchmarks and leaderboards. We achieved to drastically reduce the number of queries in these datasets with negligible performance loss, inducing considerable cost, duration, and carbon footprint savings. Overall, using only 40.7% of the datasets leads to save 26 h of inference computation and 1.8 kgCO2eq (>59% reduction), for a 0.75% relative loss on performance score. For the largest dataset in this study, we achieved 1.1% loss on performance with only 2.7% of the dataset.

Our study relies on datasets possibly reaching saturation by the latest LMs, hence future work should include more recent and complex datasets, with several metrics and domains. Moreover, we used two models that are believed to be popular and not among the most resource consuming. The study will benefit from further inclusion of resource consuming models like LRMs. Eventually, a method that allows for a real-time evaluation of the cutoff – where adding more will not help much – will be beneficial to the community.

Placing these numbers back in the whole leaderboard landscape, where dozens to hundreds of models are benchmarked, the amount of saved CO2eq could reach the order of tons and the amount of cost savings reach thousands of dollars. We also achieved significant progress towards understanding the optimal number of queries. Not enough data means too much variability and less robustness, while too many queries means a waste of resource and non-sustainable mindset.

In the era of all possible with AI but limited overall resource, this mindset should be applied when developing and evaluating industrial AI applications. Benchmarking LMs is the right approach to select the most appropriate and most sustainable model and using the most appropriate number of queries is the first step towards sustainability.

Acknowledgments

This work was supported by the AI Hub of Schneider Electric. We would like to thank Julia Peyre, Head of Strategy and Innovation, Schneider Electric, and Claude Le Pape, Fellow Data Scientist, Schneider Electric, for their support and review of the current work.

Declaration on Generative AI

The authors have not employed any Generative AI tools in the writing, review and edition of this document.

References

[1] S. Minaee, T. Mikolov, N. Nikzad, et al., Large language models: A survey, 2025. URL: https://arxiv.org/abs/2402.06196. arXiv:2402.06196.

[2] S. Luccioni, Y. Jernite, E. Strubell, Power hungry processing: Watts driving the cost of ai deployment?, in: Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency, FAccT '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 85–99. URL: https://doi.org/10.1145/3630106.3658542. doi:10.1145/3630106.3658542.

[3] S. Luccioni, B. Gamazaychikov, S. Hooker, et al., Light bulbs have energy ratings — so why can't ai chatbots?, Nature 632 (2024) 736–38. doi:10.1038/d41586-024-02680-3.

[4] C. Fourrier, N. Habib, A. Lozovskaya, K. Szafer, T. Wolf, Open llm leaderboard v2, https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard, 2024.

[5] W.-L. Chiang, L. Zheng, Y. Sheng, et al., Chatbot arena: An open platform for evaluating llms by human preference, 2024. URL: https://arxiv.org/abs/2403.04132. arXiv:2403.04132.

[6] X. Li, T. Zhang, Y. Dubois, et al., Alpacaeval: An automatic evaluator of instruction-following models, 2023. URL: https://github.com/tatsu-lab/alpaca_eval.

[7] S. J. Paech, Eq-bench: An emotional intelligence benchmark for large language models, 2024. URL: https://arxiv.org/abs/2312.06281. arXiv:2312.06281.

[8] F. Yan, H. Mao, C. C.-J. Ji, et al., Berkeley function calling leaderboard, 2024. URL: https://gorilla.cs.berkeley.edu/blogs/8_berkeley_function_calling_leaderboard.html.

[9] V. Lai, N. T. Ngo, A. P. B. Veyseh, F. Dernoncourt, T. H. Nguyen, Open multilingual llm evaluation leaderboard, 2023.

[10] N. Muennighoff, N. Tazi, L. Magne, N. Reimers, Mteb: Massive text embedding benchmark, 2023. URL: https://arxiv.org/abs/2210.07316. arXiv:2210.07316.

[11] H. Zhao, M. Li, L. Sun, T. Zhou, Bento: Benchmark task reduction with in-context transferability, 2024. URL: https://arxiv.org/abs/2410.13804. arXiv:2410.13804.

[12] F. M. Polo, L. Weber, L. Choshen, Y. Sun, G. Xu, M. Yurochkin, tinybenchmarks: evaluating llms with fewer examples, 2024. URL: https://arxiv.org/abs/2402.14992. arXiv:2402.14992.

[13] Y. Perlitz, E. Bandel, A. Gera, et al., Efficient benchmarking of language models, 2024. URL: https://arxiv.org/abs/2308.11696. arXiv:2308.11696.

[14] R. Vivek, K. Ethayarajh, D. Yang, D. Kiela, Anchor points: Benchmarking models with much fewer examples, 2024. URL: https://arxiv.org/abs/2309.08638. arXiv:2309.08638.

[15] P. Clark, I. Cowhey, O. Etzioni, et al., Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL: https://arxiv.org/abs/1803.05457. arXiv:1803.05457.

[16] K. Cobbe, V. Kosaraju, M. Bavarian, et al., Training verifiers to solve math word problems, 2021. URL: https://arxiv.org/abs/2110.14168. arXiv:2110.14168.

[17] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, Y. Choi, Hellaswag: Can a machine really finish your sentence?, 2019. URL: https://arxiv.org/abs/1905.07830. arXiv:1905.07830.

[18] D. Hendrycks, C. Burns, S. Basart, et al., Measuring massive multitask language understanding, 2021. URL: https://arxiv.org/abs/2009.03300. arXiv:2009.03300.

[19] K. Sakaguchi, R. L. Bras, C. Bhagavatula, Y. Choi, Winogrande: An adversarial winograd schema challenge at scale, 2019. URL: https://arxiv.org/abs/1907.10641. arXiv:1907.10641.

[20] S. Santurkar, E. Durmus, F. Ladhak, C. Lee, P. Liang, T. Hashimoto, Whose opinions do language models reflect?, 2023. URL: https://arxiv.org/abs/2303.17548. arXiv:2303.17548.

[21] T. B. Brown, B. Mann, N. Ryder, et al., Language models are few-shot learners, 2020. URL: https://arxiv.org/abs/2005.14165. arXiv:2005.14165.

[22] S. Shahriar, B. Lund, N. R. Mannuru, et al., Putting gpt-4o to the sword: A comprehensive evaluation of language, vision, speech, and multimodal proficiency, 2024. URL: https://arxiv.org/abs/2407.09519. arXiv:2407.09519.

[23] Microsoft, Azure openai service pricing, 2025. URL: https://azure.microsoft.com/en-us/pricing/details/cognitive-services/openai-service/.

[24] S. Rincé, A. Banse, V. Defour, Ecologits calculator, 2025. URL: https://huggingface.co/spaces/genai-impact/ecologits-calculator/.

[25] V. Satopaa, J. Albrecht, D. Irwin, B. Raghavan, Finding a 'kneedle' in a haystack: Detecting knee points in system behavior, 2011. doi:10.1109/ICDCSW.2011.20.

[26] S. Lin, J. Hilton, O. Evans, Truthfulqa: Measuring how models mimic human falsehoods, 2022. URL: https://arxiv.org/abs/2109.07958. arXiv:2109.07958.

[27] T. Sellam, D. Das, A. P. Parikh, Bleurt: Learning robust metrics for text generation, in: Proceedings of ACL, 2020.

[28] A. Analysis, Llm leaderboard, 2025. URL: https://artificialanalysis.ai/leaderboards/models/.

A. Supplementary Tables

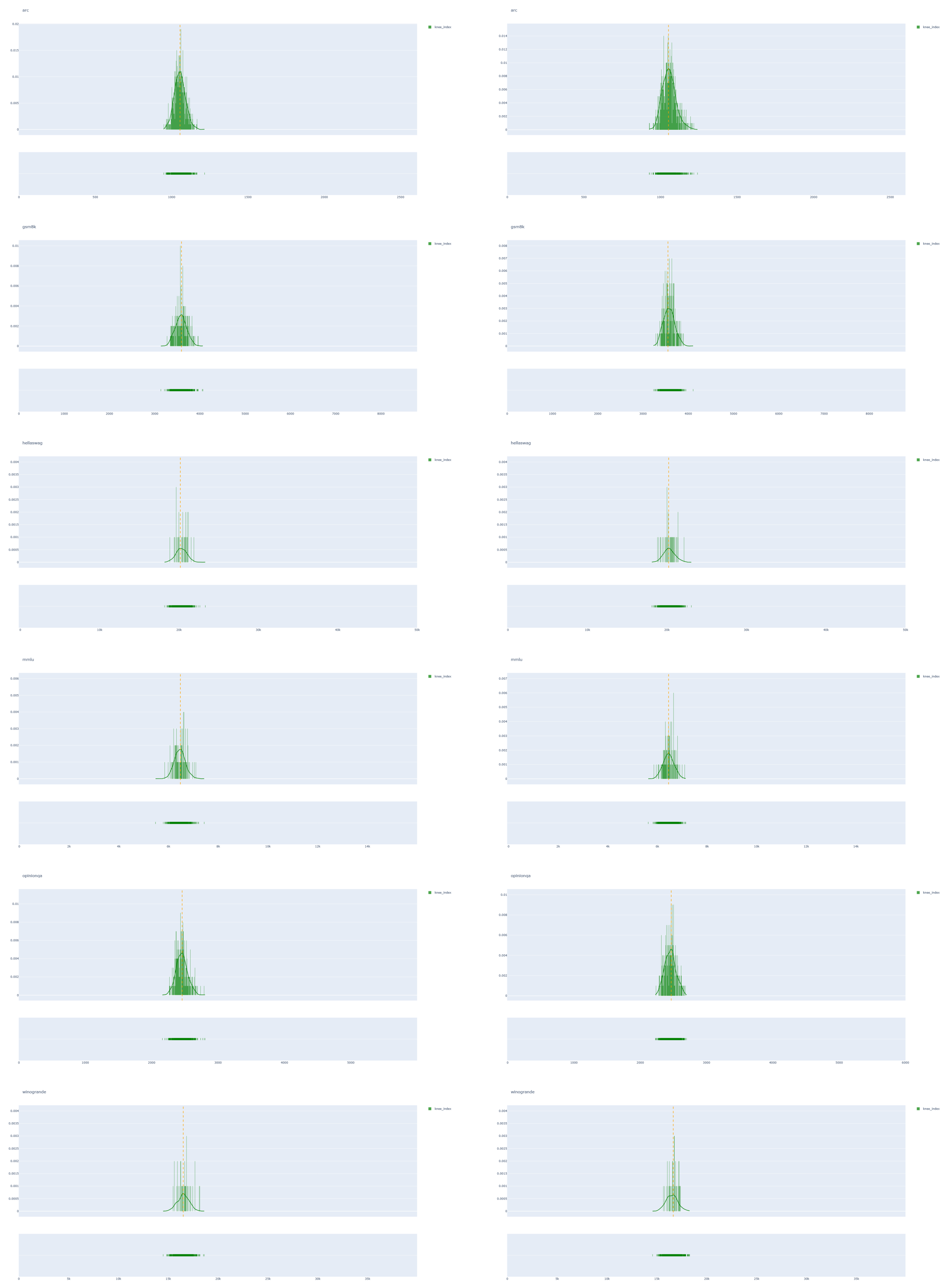
A.1
System prompts. Each system prompt starts with *"You are a useful assistant."* and ends with *"Provide your answer with the choice key in parenthesis followed by the choice sentence. Do not provide explanation for your choice."*. The following dataset-specific sections are in between.

dataset	system prompt
Arc-C	You will be told a context followed by a question, and multiple choices.
GSM8k	You will be told a grade-school level mathematical problem.
Hellaswag	You will be told a context followed by a question, and multiple choices.
MMLU	You will be told one or two contexts (scenarios, problem, social statement), followed by multiple choices.
Winogrande	You will be told a common sense context followed by sentence with a missing word, and multiple choices.
Opinionqa	You will be told a context followed by a question on political, socio-economic aspects, and multiple choices. The last choice 'Refused' is the one to choose if you judge the other choices not appropriate.

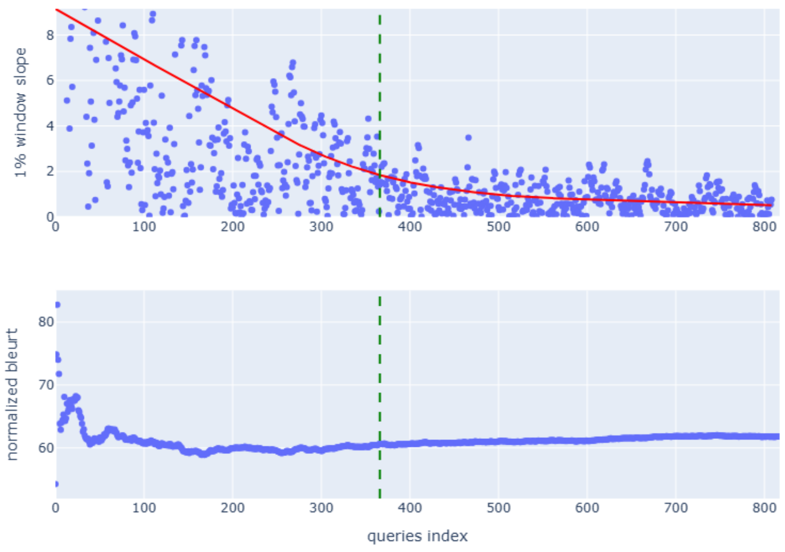
A.2
Kolmogorov-Smirnov test p-values confronting normal distribution to cutoff distribution over 1000 iterations.

model	Arc-C	GSM8k	Hellaswag	MMLU	Winogrande	Opinionqa
gpt-3.5-turbo	0.86	0.86	0.91	0.31	0.46	0.68
gpt-4o-mini	0.46	0.72	0.65	0.83	0.57	0.61

B. Supplementary Figures



A.1: Distribution of the indices detected for each iteration for kneedle method for each dataset (one dataset per line) for gpt-3.5-turbo (left column) and gpt-4o-mini (right column). The orange dashed line represents the selected cutoff (mean).



A.2: The tracking of the slopes (top, blue dots) for one iteration of randomizing the TruthfulQA queries, computed over each successive 8-queries frames (1% of 817 queries), with lowess smoothing trendline (red). Tracking of the performance score (bottom) along the same queries, reaching the final performance score (61.84) at last. The kneedle algorithm identifies here the plateau root (dashed green line) at the 366th query.