OckBench: Tokens are Not to Be Multiplied without Necessity

Anonymous Author(s)
Affiliation
Address
email

Abstract

Large language models (LLMs) such as GPT-4, Claude 3, and the Gemini series have pushed the frontier of automated reasoning and code generation. Yet, prevailing benchmarks emphasize accuracy and output quality, neglecting a critical dimension: decoding token efficiency. In real systems, the difference between generating 10K tokens vs 100K tokens is nontrivial in latency, cost, and energy. In our work, we introduce OckBench, the first model-agnostic, hardware-agnostic benchmark that jointly measures accuracy and decoding token count for reasoning and coding tasks. Through experiments comparing multiple open- and closed-source models, we uncover that many models with comparable accuracy differ wildly in token consumption, revealing that efficiency variance is a neglected but significant axis of differentiation. We further demonstrate Pareto frontiers over the accuracy–efficiency plane and argue for an evaluation paradigm shift: we should no longer treat tokens as "free" to multiply. OckBench provides a unified platform for measuring, comparing, and guiding research in token-efficient reasoning.

1 Introduction

"Entities must not be multiplied beyond necessity."

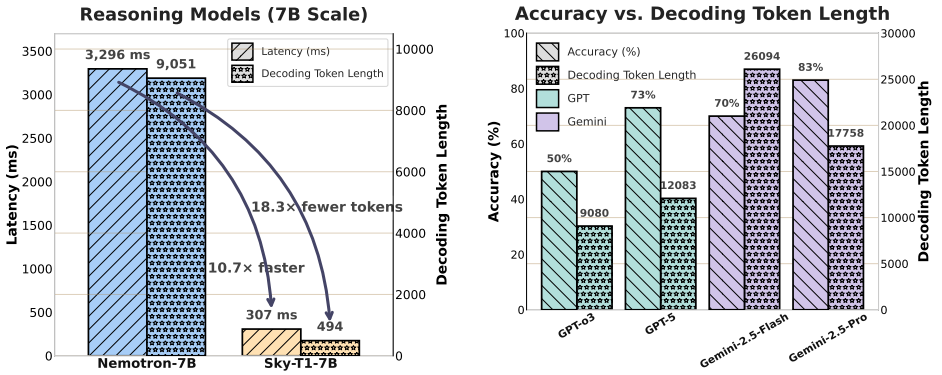
— The Principle of Ockham's Razor

Large Language Models (LLMs) such as GPT-4, Claude 3, and Gemini have demonstrated remarkable capabilities in complex problem-solving, largely attributed to their advanced reasoning abilities. Techniques like Chain of Thought (CoT) prompting and self-reflection have become central to this success, enabling models to perform step-by-step deductions for tasks requiring deep knowledge and logical rigor, such as advanced mathematics and programming challenges. As the industry increasingly emphasizes this "long decoding" mode, the computational cost associated with these reasoning processes has grown significantly. For instance, public reports indicate that frontier models may require over **ten hours** to solve just six mathematical problems [1], and in coding competitions, some difficult problems take models more than **two hours** to complete [2]. These examples illustrate a broader issue: while the community often celebrates model accuracy on challenging tasks, the substantial time and computational costs involved in achieving such results receive far less discussion.

While LLM evaluation and comparison have become increasingly important, most evaluations focus primarily on the accuracy but the efficiency of generation is less discussed. For example, HELM [3], LM-Eval [4], and the LMSYS Chatbot Arena [5] rank almost mostly on task accuracy. This suggests that the number of decoding tokens, a model- and hardware-agnostic metric, plays a major role in determining practical efficiency across tasks.

To address this overlooked dimension of reasoning efficiency, we introduce a new evaluation perspective centered on intrinsic token efficiency. Our contributions are summarized as follows:

Submitted to NeurIPS 2025 Workshop on Efficient Reasoning. Do not distribute.



(a) Reasoning Efficiency Comparison among Similar Size Models

(b) Closed-source Model Reasoning Efficiency and Accuracy Comparison

Figure 1: (1a) shows that even similar-sized models can exhibit a 10.7× difference in reasoning time due to varying decoding token counts. (1b) shows that frontier closed-source models have comparable accuracy but vary significantly in reasoning efficiency.

- **Model-Agnostic Efficiency Metric.** We formalize decoding token count as an intrinsic, hardware- and system-independent efficiency metric, complementing accuracy to provide a more holistic view of model performance and guiding both model design and training.
- **Efficiency-Accuracy Aware Benchmark.** We propose **OckBench**, the first unified benchmark specifically designed to evaluate the efficiency of an LLM's reasoning process by measuring decoding token consumption alongside accuracy.
- **Empirical Efficiency-Accuracy Trade-offs.** We conduct experiments across multiple open- and closed-source models, illustrating their distribution on an accuracy–efficiency Pareto frontier and revealing substantial practical trade-offs.

2 Toward a Unified Model-Agnostic Reasoning Efficiency Framework

2.1 Practical Cost of LLMs

As model sizes scale and real-time serving requirements become ubiquitous, the inference budget for large language models (LLMs) has emerged as a critical deployment bottleneck. Each additional decoding token incurs non-trivial latency, energy consumption, and monetary cost. Indeed, LLM service providers commonly report billing in units of millions of output tokens, highlighting that **output token generation** now dominates operational expenditures [6]. Meanwhile, empirical analysis by Epoch AI shows that the response lengths of reasoning-capable models have been growing at roughly **5× per year**, whereas those of non-reasoning models have grown at around **2.2× per year** [7]. This divergence underscores that as reasoning capabilities advance, so too does the hidden cost of "thinking" in token form.

2.2 Invisible Inefficiency in Current Optimization and Evaluation

Most existing efficiency efforts focus on orthogonal components such as weight compression, quantization, hardware acceleration, or system scheduling [8, 9]. While there are studies on efficient reasoning or decoding optimization that aim to reduce generated tokens [10, 11, 12], these approaches typically do not provide a unified benchmark that enables fair comparison of reasoning efficiency across different models and task domains.

Meanwhile, mainstream evaluation frameworks primarily emphasize output quality—accuracy, robustness, and fairness—while paying less attention to the number of reasoning tokens generated. Other efficiency-oriented frameworks (e.g., MLPerf) measure system- or hardware-level performance (throughput, latency, CO_2 emissions) [13]. These metrics are informative for deployment infrastructure, but they do not directly reveal the intrinsic reasoning efficiency of the model itself.

To provide clearer guidance for token-efficiency research and to evaluate reasoning efficiency more intrinsically, we adopt **decoding token count** (on a fixed task under a fixed decoding setting) as our core efficiency metric. Building on this metric, we present **OckBench**, the first unified benchmark that

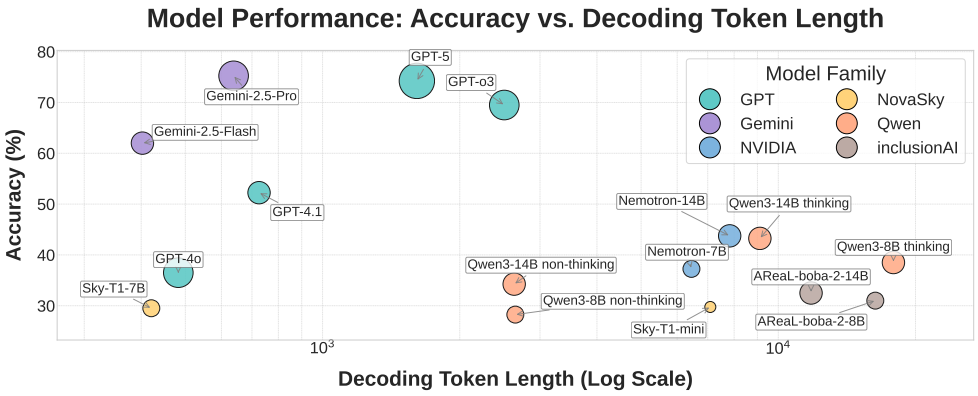


Figure 2: Reasoning Efficiency Comparison Among 16 Models.

is accuracy-efficiency aware, model-agnostic, and hardware-agnostic, enabling fair and reproducible comparisons of reasoning efficiency across LLMs.

3 OckBench Benchmark

3.1 Benchmark Composition

Our benchmark, OckBench, is structured to test LLMs' **reasoning efficiency** across two complementary domains: math problems solving and coding skills.

Mathematics and Reasoning Tasks. We adopt GSM8K[14], AIME24, and AIME25 as core reasoning benchmarks. To better expose token-efficiency differences, we select the top 200 questions that exhibit high variance in decoding token usage among baseline models.

Software Engineering Tasks. For the coding domain, we build a lightweight variant of MBPP [15], supplemented by 200 carefully curated real-world coding problems using the same criterion as the math dataset. These coding tasks cover algorithmic challenges, code transformation, debugging, and small-scale project tasks.

3.2 Question Combination.

Our decoding token variance based selection balances difficulty diversity and token-variance sensitivity. We aim to include questions that are not trivially solved (to avoid floor effects) nor overwhelmingly hard (to avoid zero accuracy), while also maximizing the spread in decoding token usage across models. This design helps the benchmark emphasize efficiency contrast among models, rather than merely ranking by accuracy. This helps to design and evaluate more token efficient and robust models.

4 Experiments

4.1 Setup

We select and evaluate a set of both open- and closed-source models with varying parameter sizes (see Model List in subsection 4.2). We gather each model's decoding token count and accuracy on two domains: a mathematics dataset (GSM8K, AIME '24 and AIME '25) and a coding dataset (MBPP [15]).

From the combined results, we then select the top 200 instances exhibiting the greatest variance in decoding token count across models. This is a core methodological choice for OckBench. A problem where all models use a similar number of tokens tells us nothing about efficiency, even if accuracies differ. By selecting for high variance, we are filtering for the specific instances that force models to reveal their true reasoning efficiency and best exemplify the accuracy-efficiency trade-offs this paper investigates. This ensures our benchmark is composed of problems where token count is a decisive and high-contrast metric.

4.2 Models and Tasks

The following models were included in the analysis:

- **Commercial Models:** GPT-5 [16], Gemini 2.5 Pro [17], GPT-o3 [18], Gemini 2.5 Flash [19], GPT-4.1 [20], and GPT-4o [21].
- **Open-Source Models:** AceReason-Nemotron (14B, 7B) [22], Qwen3-(14B, 8B, 4B, each with "thinking" and "non-thinking" variants) [23], inclusionAI AReaL-boba-2 (14B, 8B) [24, 25], and NovaSky-AI Sky-T1 (7B, mini) [26].

OckBench-Math. The first benchmark is evaluated models on the top 200 most challenging problems from the gsm8k and AIME24/25 dataset to test their mathematical problem-solving abilities.

Code Generation. The models were also evaluated on a set of 200 variant coding problems from MBPP dataset to assess their programming and logical reasoning capabilities.

Table 1: Overall Performance Rankings on **OckBench-Math**. Ranked by Reasoning Efficiency (#Tokens / Acc). *Sky-T1-7B demonstrates superior performance because

Model	Category	#Tokens	Accuracy (%)	Reasoning Efficiency
GPT-4o	Commercial	495	35	14.1
GPT-4.1	Commercial	872	59	14.9
Sky-T1-7B	Open-Source	556	33	17.1
GPT-5	Commercial	2,336	73	32.2
GPT-o3	Commercial	2,347	64	36.8
Gemini-2.5 Flash	Commercial	4,777	66	72.6
Gemini-2.5 Pro	Commercial	5,198	68	76.2
Qwen3-14B (non-thinking)	Open-Source	3,010	33	92.0
Qwen3-4B (non-thinking)	Open-Source	3,494	30	118.4
Qwen3-8B (non-thinking)	Open-Source	3,692	30	124.1
Nemotron-14B	Open-Source	5,540	40	139.4
Sky-T1-mini	Open-Source	6,657	33	204.8
Qwen3-14B (thinking)	Open-Source	8,190	40	206.0
Nemotron-7B	Open-Source	8,895	35	254.2
AReaL-boba-2-14B	Open-Source	10,439	38	278.4
AReaL-boba-2-8B	Open-Source	17,038	37	457.4
Qwen3-8B (thinking)	Open-Source	20,440	38	541.5
Qwen3-4B (thinking)	Open-Source	24,025	37	649.3

The comprehensive results for mathematical problems are presented in Table 1. The models are ranked based on their accuracy. The the average decoding token length, which serves as a measure of verbosity and computational cost.

Table 2 presents the "pass at one" rate, which measures the percentage of problems solved correctly on the first attempt, alongside the average number of generated tokens.

4.3 Main Result

Figure 2 illustrates the comparison of accuracy versus decoding token count for models in OckBench, with a comprehensive comparison shown in Table Table 1. Our experiments shows that there is a **significant reasoning efficiency gap** between commercial (closed-source) and open-source models, details below:

Commercial models demonstrated superior performance, with an average accuracy of 60.8%. GPT-5 achieved the highest accuracy at 73%. Notably, there is a wide variance in token efficiency among commercial models; while GPT-5 was highly accurate and concise (2,336 tokens), Gemini-2.5 Pro required over two times as many tokens (5,198) to achieve a slightly lower accuracy. GPT-4o stands out as the most token-efficient commercial model, though its accuracy was lower than the top performers.

Table 2: Overall Performance Rankings on Top 200 Coding Problems

Model	Category	#Tokens	Accuracy (%)	Reasoning Efficiency
GPT-4o	Commercial	491	38	12.9
Sky-T1-7B	Open-Source	348	23	15.1
GPT-4.1	Commercial	782	47	16.6
GPT-5	Commercial	1,436	75	19.1
Gemini 2.5 Pro	Commercial	1,798	77	23.4
Gemini 2.5 Flash	Commercial	2,346	60	39.1
GPT-o3	Commercial	3,001	71	42.3
Qwen3-4B (non-thinking)	Open-Source	1,700	28	60.7
Qwen3-14B (non-thinking)	Open-Source	2,413	35	68.9
Qwen3-8B (non-thinking)	Open-Source	2,098	27	77.7
Nemotron-14B	Open-Source	9,840	46	213.9
Qwen3-14B (thinking)	Open-Source	10,498	48	218.7
Sky-T1-mini	Open-Source	5,603	24	233.5
Qwen3-8B (thinking)	Open-Source	11,738	41	286.3
Qwen3-4B (thinking)	Open-Source	12,563	39	322.1
Nemotron-7B	Open-Source	12,895	40	322.4
AReaL-boba-2-14B	Open-Source	12,648	32	395.3
AReaL-boba-2-8B	Open-Source	14,537	31	468.9

Open-source models had a lower average accuracy of 35.3%. NVIDIA's AceReason-Nemotron-14B and Qwen's Qwen3-14B were the top performer in this category (40% accuracy). A clear trend is visible where "thinking" variants of the Qwen models, which likely use more extensive chain-of-thought processing, produced substantially higher token counts compared to their "non-thinking" counterparts, without a proportional increase in accuracy. The NovaSky-AI Sky-T1-7B model provided a good balance of performance and efficiency within the open-source group, achieving a respectable accuracy with a low average token count, comparable to the most efficient commercial models.

5 Conclusion

In this paper, we introduced OckBench, a unified benchmark that brings reasoning efficiency, measured via decoding token length alongside accuracy in the evaluation of large language models (LLMs). Through experiments comparing both open- and closed-source models across mathematics and coding domains, we found that models with comparable accuracy can differ substantially in token consumption. For example, among commercial models, one high-accuracy model required over $4\times$ the tokens of another to reach a slightly lower accuracy. Among open-source models, we observed that variants optimized for more extensive chain-of-thought reasoning often consumed far more tokens without proportional accuracy gains. We also find that small models could be inefficient compared with bigger models given the same reasoning task due to different decoding token length.

These findings highlight that token efficiency is a meaningful axis of differentiation, especially in deployment contexts where latency, computation, and cost matter. By adopting a metric that is model- and hardware-agnostic, OckBench provides a reproducible and fair platform for comparing the accuracy–efficiency trade-off of reasoning models. We hope this benchmark will guide the community toward designing models that not only get things right, but do so with leaner, more efficient reasoning. Future work under this framework may explore dynamic reasoning budgets, early-exit mechanisms, token-pruning strategies, and further evaluation on additional domains and real-world workloads.

References

[1] OpenAI. Learning to reason with llms, September 2024.

[2] Thang Luong and Edward Lockhart. Advanced version of gemini with deep think officially achieves gold-medal standard at the international mathematical olympiad, July 2025.

[3] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Alexander Cosgrove, Christopher D Manning, Christopher Re, Diana Acosta-Navas, Drew Arad Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue WANG, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Andrew Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023. Featured Certification, Expert Certification.

[4] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024.

[5] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating llms by human preference, 2024.

[6] Rohail Saleem. Duolingo allegedly tops a list of openai's top 30 customers by token consumption, October 2025.

[7] Luke Emberson, Ben Cottier, Josh You, Tom Adamczewski, and Jean-Stanislas Denain. Llm responses to benchmark questions are getting longer over time, 2025. Accessed: 2025-10-19.

[8] Yujun Lin, Haotian Tang, Shang Yang, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, and Song Han. Qserve: W4a8kv4 quantization and system co-design for efficient llm serving, 2025.

[9] Hao Kang, Qingru Zhang, Souvik Kundu, Geonhwa Jeong, Zaoxing Liu, Tushar Krishna, and Tuo Zhao. Gear: An efficient kv cache compression recipe for near-lossless generative inference of llm, 2024.

[10] Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. Token-budget-aware llm reasoning, 2025.

[11] Junlin Wang, Siddhartha Jain, Dejiao Zhang, Baishakhi Ray, Varun Kumar, and Ben Athiwaratkun. Reasoning in token economies: Budget-aware evaluation of LLM reasoning strategies. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19916–19939, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[12] Junhong Lin, Xinyue Zeng, Jie Zhu, Song Wang, Julian Shun, Jun Wu, and Dawei Zhou. Plan and budget: Effective and efficient test-time scaling on large language model reasoning, 2025.

[13] Vijay Janapa Reddi, Christine Cheng, David Kanter, Peter Mattson, Guenther Schmuelling, Carole-Jean Wu, Brian Anderson, Maximilien Breughe, Mark Charlebois, William Chou, Ramesh Chukka, Cody Coleman, Sam Davis, Pan Deng, Greg Diamos, Jared Duke, Dave Fick, J. Scott Gardner, Itay Hubara, Sachin Idgunji, Thomas B. Jablin, Jeff Jiao, Tom St. John, Pankaj Kanwar, David Lee, Jeffery Liao, Anton Lokhmotov, Francisco Massa, Peng Meng, Paulius Micikevicius, Colin Osborne, Gennady Pekhimenko, Arun Tejusve Raghunath Rajan, Dilip Sequeira, Ashish Sirasao, Fei Sun, Hanlin Tang, Michael Thomson, Frank Wei, Ephrem Wu, Lingjie Xu, Koichi Yamada, Bing Yu, George Yuan, Aaron Zhong, Peizhao Zhang, and Yuchen Zhou. Mlperf inference benchmark, 2019.

[14] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

[15] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

[16] OpenAI. Introducing gpt-5, 2025. Accessed: 2025-10-31.

[17] Google DeepMind. Introducing gemini 2.5 pro: Advanced reasoning model, 2025. Accessed: 2025-10-31.

[18] OpenAI. Introducing openai o3, 2025. Accessed: 2025-10-31.

[19] Google DeepMind. Introducing gemini 2.5 flash: Fast and efficient reasoning model, 2025. Accessed: 2025-10-31.

[20] OpenAI. Introducing gpt-4.1, 2025. Accessed: 2025-10-31.

[21] OpenAI. Hello gpt-4o ("o" for "omni"), 2024. Accessed: 2025-10-31.

[22] Yang Chen, Zhuolin Yang, Zihan Liu, Chankyu Lee, Peng Xu, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Acereason-nemotron: Advancing math and code reasoning through reinforcement learning. *arXiv preprint arXiv:2505.16400*, 2025.

[23] Qwen Team. Qwen3 technical report, 2025.

[24] Wei Fu, Jiaxuan Gao, Xujie Shen, Chen Zhu, Zhiyu Mei, Chuyi He, Shusheng Xu, Guo Wei, Jun Mei, Jiashu Wang, Tongkai Yang, Binhang Yuan, and Yi Wu. Areal: A large-scale asynchronous reinforcement learning system for language reasoning, 2025.

[25] Zhiyu Mei, Wei Fu, Kaiwei Li, Guangju Wang, Huanchen Zhang, and Yi Wu. Real: Efficient rlhf training of large language models with parameter reallocation. In *Proceedings of the Eighth Conference on Machine Learning and Systems, MLSys 2025, Santa Clara, CA, USA, May 12-15, 2025*. mlsys.org, 2025.

[26] NovaSky Team. Unlocking the potential of reinforcement learning in improving reasoning models. https://novasky-ai.github.io/posts/sky-t1-7b, 2025. Accessed: 2025-02-13.

A Experiments Results and Details

This appendix details the performance analysis of 18 different AI models across two challenging benchmarks: advanced mathematical reasoning and code generation. The objective was to evaluate and compare the accuracy, token consumption, and overall efficiency of these models. The models were categorized into two groups: commercial and open-source.