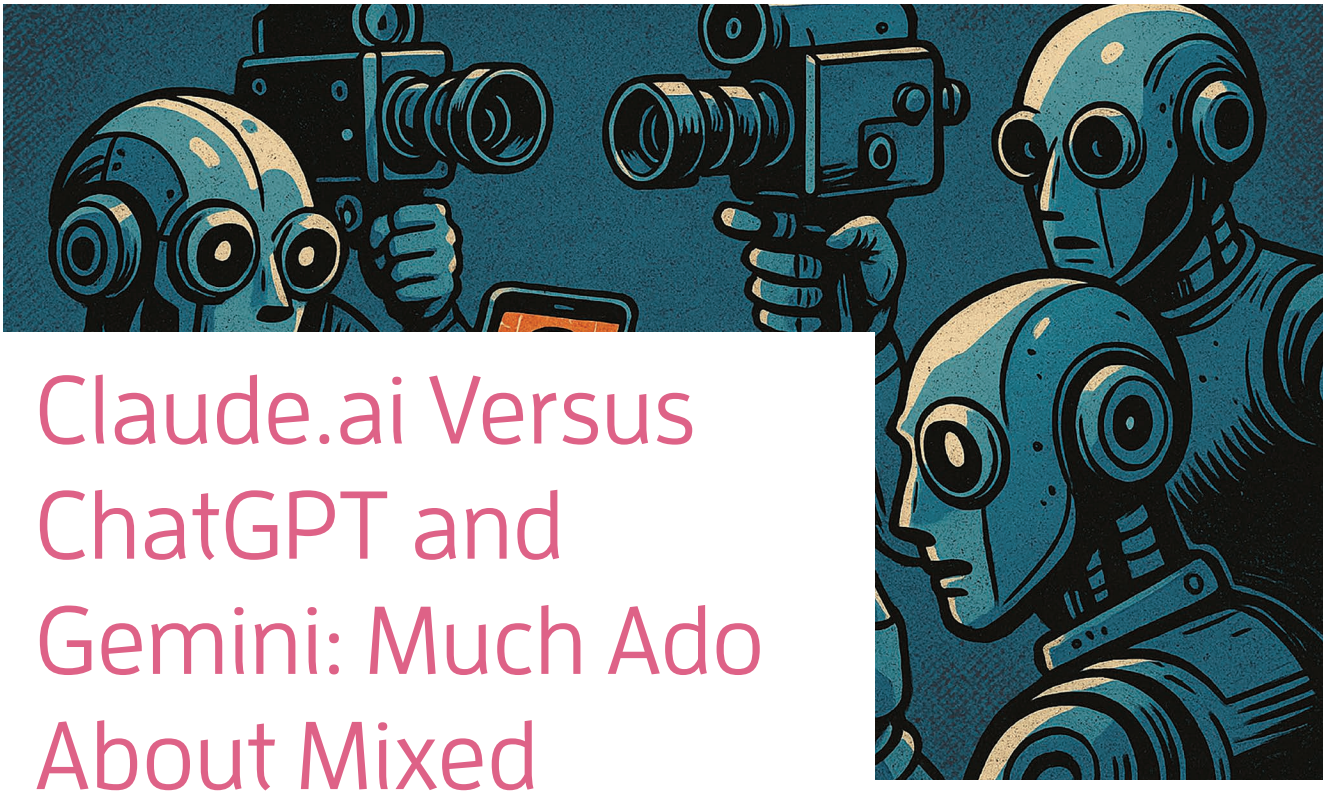

GRAPHICS GENERATED USING CHATGPT 5.0

Claude.ai Versus ChatGPT and Gemini: Much Ado About Mixed

Sorin Faibish, Life Senior Member, IEEE

A real-world comparison of Anthropic's Claude, OpenAI's ChatGPT-4o, and Google's Gemini reveals key differences in speed, consistency, and user experience, highlighting the impact of streaming versus crawling.

This comparative study evaluates the real-time to answer (RTTA) performance of Claude, ChatGPT-4o, and Gemini[6] across 25 practical, structured workloads to inform enterprise and technical user decisions.

Key findings are as follows:

- ChatGPT-4o is on average 9% faster than Claude, with consistent latency and responsive streaming output.
- Gemini[7] is on average 12% faster than Claude, achieving high performance on certain technical tasks but showing latency spikes in translation and prompt-heavy queries due to real-time web crawling.
- Architectural designs significantly shape user experiences: Claude and Gemini share similarities in pipeline-based output, while ChatGPT benefits from immediate streaming token generation.
- Workloads involving translation, multistage reasoning, and creative generation exhibit the most variance, underscoring the need for task-aligned model selection.
- Even minor RTTA differences compound in scaled environments, affecting productivity in enterprise pipelines and developer workflows.

Implications are as follows:

- *Use ChatGPT-4o* when consistent interaction speed and responsiveness are critical.

Digital Object Identifier 10.1109/MC.2025.3587774
Date of current version: 22 December 2025

EDITOR **MICHAEL ZYDA**
University of Southern California;
zyda@mikezyda.com

- *Use Gemini* when high-speed retrieval with up-to-date information is prioritized, with the understanding of potential latency variability.
- *Claude* remains a stable reference for batch reasoning tasks but is comparatively slower in end-to-end RTTA.

The results highlight that the choice of large language model (LLM) is context dependent, and RTTA benchmarking can guide deployment strategies for engineering, content generation, and knowledge workflows.

The rapid evolution of LLMs has transformed how developers and enterprises approach knowledge workflows, coding assistance, and automated content generation. However, while much attention has been paid to output quality, factual accuracy, and cost, the user experience is also heavily impacted by RTTA, which defines the latency between prompt submission and full output delivery.

Following the call for rigorous LLM evaluation in Michael Zyda's "Much Ado About DeepSeek"[1] and expanded in "Much Ado About ChatGPT Versus DeepSeek,"[2,3,4] this article benchmarks Anthropic's Claude against OpenAI's ChatGPT-4o and Google's Gemini to quantify practical differences in speed and consistency across diverse, real-world workloads. We aim to supplement architectural discussions with empirical timing data that capture the user-facing realities of these models under common developer and enterprise scenarios.

The evaluation focuses on structured workloads, including technical queries (for example, CUDA and S3 usage), cybersecurity, translation tasks, infrastructure discussions, and creative generation. Using Claude as the baseline, we measure relative RTTA ratios to analyze which models deliver faster output and under what conditions. By pairing these results with an analysis of architectural approaches, streaming outputs, pipeline-based generation, and real-time crawling, we clarify where each model excels and where it faces limitations.

In doing so, this study provides a practical guide for engineers, researchers, and enterprise technology leaders seeking to align LLM deployments with workflow latency requirements while accounting for architectural tradeoffs that shape the user experience.

THE EXPERIMENTAL SETUP AND METHODOLOGY

Workload composition

The benchmark set was carefully curated to reflect the diversity of real-world user prompts that enterprise engineers, researchers, and advanced users submit to LLMs. Workloads spanned the following:

- *Technical infrastructure queries*: These include GPU clusters, CUDA in HPC, S3, file systems, and surface mount technology.
- *Applied generative AI (GenAI) tasks*: These cover retrieval-augmented generation (RAG) studies, GenAI for specialized industries (for example, food and supply chain design), and cybersecurity incident response planning.
- *Code and scripting tasks*: These use prompts that request snippet generation, code explanations, or infrastructure-as-code workflows.
- *Creative and analytical generation*: These include resume drafting, business deal analysis, and scenario-based planning.
- *Translation and linguistic work*: These include "translate to French" and comparative linguistic analysis of French and English structures.
- *Exploratory research tasks*: These use open-ended prompts requiring summarization of new technology or definition-based fact retrieval.

This composition ensured balanced representation of prompt types that impact practical RTTA while providing insights into how each LLM manages structured, technical, and creative workloads.

Workload design

A total of 25 workloads were initially tested. These covered the following:

- technical knowledge (for example, CUDA usage and GPU cluster builds)
- applied AI (for example, GenAI in food and RAG studies)
- creative generation (for example, poetry and resume writing)
- code and infrastructure (for example, MPI versus OpenMPI, S3, and file systems)
- language translation and comparative linguistics
- cybersecurity and cloud architecture queries.

From a broader set of workloads, the most relevant 25 were selected for the final report to balance RTTA performance and ensure diverse domain coverage.

Measurement approach

The measurement of RTTA was conducted using a manual stopwatch-based timing method:

- *Timing start*: When the prompt was submitted (enter/send key pressed).
- *Timing end*: When the last token of the LLM's output was fully rendered in the user interface.

This approach was chosen over token-stream monitoring or API-based capture to reflect the *true user experience latency*, including back-end processing, streaming (where applicable), and interface rendering delays.

Key considerations

- Prompt structures were standardized across models to ensure fair timing comparisons.
- Each test was conducted under stable, high-bandwidth network conditions to minimize client-side delays.
- Testing was performed on *paid, latest stable versions of each LLM* to reflect the best-available performance.

Using Claude as the reference baseline allowed the direct calculation of *relative RTTA ratios*, where a ratio >1 indicates the comparator model was faster for the given workload.

Measurement strategy

To ensure consistency, reliability, and interpretability of RTTA measurements, the following strategy was employed:

- *Repetition*: Each workload was tested three times per model on different days and times to capture variations due to the server load, network conditions, and dynamic model updates.
- *Normalization*: Prompt lengths and response limits were harmonized across models where possible, ensuring that differences in verbosity did not skew timing.
- *Context mode*: For Gemini, "deep research" mode was enabled to allow real-time crawling, reflecting its intended architecture for up-to-date factual retrieval.
- *Averaging*: The three timing results for each workload–model pair were averaged to compute the final RTTA used in ratio calculations.
- *Ratio calculation*: RTTA ratios were calculated by dividing Claude's timing by ChatGPT's and Gemini's timing for the same workload, yielding intuitive "speedup" values for each comparison.

This measurement strategy allowed us to align benchmarking with practical, user-visible latencies, providing enterprise decision makers and technical users with actionable insights into expected performance in real-world deployment conditions.

THE RESULTS: A NUMERIC SNAPSHOT

The curated 25-workload benchmark highlights consistent but nuanced performance differences among Claude, ChatGPT-4o, and Gemini, which are presented in Table 1. Using Claude as the baseline, the results were as follows:

- ChatGPT-4o averaged a 1.09× RTTA ratio, indicating a 9% faster overall response.
- Gemini averaged a 1.12× RTTA ratio, translating to a 12% faster average RTTA.

However, the data revealed substantial variance across specific workloads, as follows:

- Gemini achieved exceptional speedups on infrastructure and technical retrieval tasks (for example, 2.14× on "download public LLM") due to efficient back-end retrieval and occasional effective real-time retrieval.
- Gemini lagged significantly in translation and linguistics workloads (for example, 0.34× on "translate to French"), indicating architecture-related delays.
- ChatGPT-4o exhibited stable, moderate improvements across nearly all workloads, with particularly strong performance in coding and general technical Q&A tasks.

This snapshot emphasizes that *RTTA advantages are not uniform* and that architectural factors (streaming versus real-time crawling, inference pipelines, and prompt handling) heavily influence performance outcomes.

ANALYSIS

Architectural implications

Claude's inference structure prioritizes batch reasoning stability, often providing comprehensive answers without needing web context, which results in competitive RTTA on structured technical workloads but lags in absolute speed where Gemini and ChatGPT excel.

ChatGPT-4o leverages a streaming token generation architecture, providing users with near-instant feedback while generating outputs. This architecture suits interactive workflows, coding support, and stepwise generation tasks where perceived responsiveness is as important as absolute speed.

Gemini's design leverages live-data crawling, which, while providing freshness, incurs startup latency[3] due to site selection and real-time retrieval. This tradeoff creates peaks of performance (in retrieval-heavy prompts) but valleys in latency-sensitive tasks.

Workload sensitivity

Workloads requiring multistage reasoning (for example, cybersecurity incident planning and infrastructure architecture) showed closer RTTA performance across models, while translation, high-fact density, and creative generation exhibited the widest variance.

Gemini performed best in scenarios demanding fresh context, whereas ChatGPT-4o consistently provided stable speeds on code and structured Q&A. Claude demonstrated steady performance, confirming its reliability for workflows tolerating marginally higher latency.

- Gemini shows high peaks (for example, "download public

LLM" at 2.14×) but also low troughs (for example, 0.34× in translation), reflecting its dependence on live web queries and architecture.
- ChatGPT offers more consistent gains, outperforming Claude in technical and code workloads while matching or trailing in translation.
- Both outperform Claude overall but not uniformly across all tasks.
- Tasks like "GenAI in food" and "translate to French" are slow across all LLMs, indicating prompt complexity or low corpus optimization.

INTERPRETING THE WIDE RTTA VARIANCE

Figure 1 illustrates the primary contributors to RTTA variance among Claude.ai, ChatGPT-4o, and Gemini, highlighting how architectural design decisions translate directly into measurable latency differences.

Web crawl delay (Gemini)

Gemini's architecture integrates real-time web crawling[3] for fresh factual grounding, introducing unpredictable startup delays due to HTTP request latencies, variable site availability, and live content parsing. While this ensures up-to-date responses, it creates the largest variability in RTTA, particularly in retrieval-heavy tasks.

Streaming output (ChatGPT-4o)

ChatGPT-4o utilizes token streaming, outputting partial results as generation progresses, significantly reducing perceived latency even on longer responses. This streaming design minimizes output delay variance and makes ChatGPT-4o the most consistent performer across workloads as users receive immediate feedback while generation continues.

Prompt complexity sensitivity

All three models exhibit some RTTA variability based on prompt structure and complexity. Multistage reasoning, multipart instructions,[10] and translation requests lead to longer planning phases in Claude and Gemini, while ChatGPT's optimized context window handling mitigates some variability but still shows modest increases under complex generation tasks.

Hardware and infrastructure variance

LLMs hosted on scalable infrastructure are affected by server load, GPU allocation, and back-end queuing, contributing to minor but noticeable RTTA variance. ChatGPT-4o and Claude generally manage these variations well, while Gemini's dependency

TABLE 1. An RTTA comparison snapshot (25 selected workloads + average).

Tested workload	ChatGPT/Claude RT	Gemini RT/Claude RT
Download public LLM	1.29	2.14
Surface mount technology	0.85	0.51
Run LLMs on local server	1.34	1.29
CUDA usage in HPC	1.79	1.90
CO_2 emission facts	1.04	1.09
Supply chain design	1.47	1.47
Amazon contact centers	1.79	1.79
Immersion cooling manufacturing	0.97	1.49
Use of LLM for coding	1.30	1.30
Define cooling technology	1.49	1.49
GenAI in food applications	0.53	0.53
What are foundational models?	1.10	0.51
Build contact center	1.79	1.79
Long-range drone surveillance	1.79	1.79
Add private data to local LLM	1.29	0.56
E-mail analysis	0.85	0.85
Compare French and English	0.50	0.34
Examples of L1 hacks	0.70	0.70
Business deal analysis	1.04	1.04
Cyberincident response	1.42	1.42
What is S3?	0.59	0.59
Human risk management study	0.62	0.62
RAG study	0.51	1.57
File systems in arrays	0.82	0.82
Translate to French	0.34	0.34
Average RTTA ratios	1.09	1.12

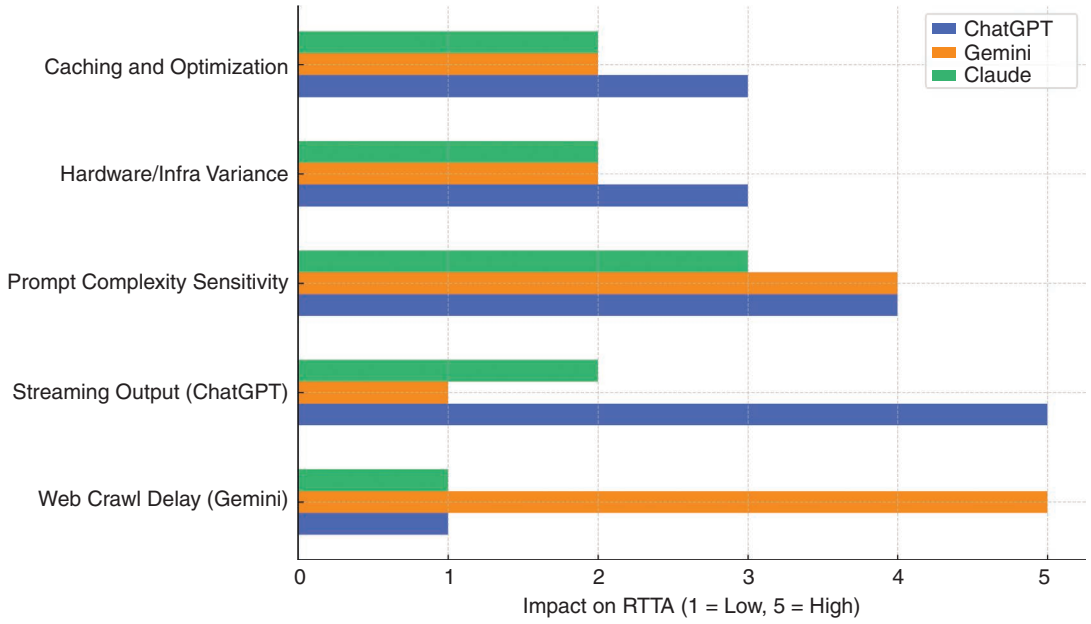

FIGURE 1. Contributors to RTTA variance among Claude.ai, ChatGPT-4o, and Gemini.

on external retrieval chains amplifies infrastructure variance when network conditions fluctuate.

Caching and optimization

Claude and ChatGPT-4o leverage prompt caching and model optimization techniques that stabilize RTTA on repetitive or commonly structured queries. Gemini benefits less from caching due to its emphasis on live retrieval, leading to additional variability when prompt data must be fetched in real time.

Overall, Gemini's variability is driven by its commitment to real-time web data freshness, while ChatGPT-4o's low variability stems from its streaming[11] architecture and consistent internal caching. Claude's variance remains moderate, reflecting its focus on structured, complete generation with less emphasis on live data retrieval.

These factors confirm that architectural design choices, rather than raw model speed, are the primary drivers of RTTA variability across LLMs,[8], informing strategic decisions in selecting and deploying these models for latency-sensitive enterprise and developer workflows.

This comparative RTTA study of Claude, ChatGPT-4o, and Gemini reveals the following:

- ChatGPT-4o is the most balanced performer, offering low-latency, consistent response speeds suitable for developer-centric and user-interactive tasks.
- Gemini demonstrates peak RTTA performance in retrieval-heavy workloads but suffers latency penalties in translation and nontrivial reasoning workloads, making it best suited for real-time data freshness use cases.
- Claude remains a reliable, stable reference, favoring structured, predictable outputs while trading off speed for thoroughness.

For enterprise and developer decision makers, the choice of LLM should align with the following:

- interaction requirements (ChatGPT-4o)
- need for real-time factuality (Gemini)
- stable, structured batch processing (Claude).

This study confirms that even small RTTA differences compound at scale in enterprise pipelines, affecting workflow efficiency and user satisfaction. Future work should explore cost–performance tradeoffs, GPU utilization impacts, and advanced streaming strategies to optimize LLM deployments in high-volume environments.

Note that I used all three LLM solutions to help me write the article based on my directives. It could be improved, but I am wondering if it is not fancier to use the LLM solutions to write the article.

ACKNOWLEDGMENT

The author thanks the developers and support teams of Gemini, ChatGPT-4o, and Claude.ai for enabling open access to their platforms, which made the comparative study possible. The author especially thanks Michael Zyda for his inspiring column, "Much Ado About DeepSeek" in *Computer*, which motivated this column and provided a thoughtful foundation for framing the discussions on LLMs.

REFERENCES

1. M. Zyda, "Much ado about DeepSeek ...," *Computer*, vol. 58, no. 5, pp. 78–81, May 2025, doi: 10.1109/MC.2025.3541112.
2. S. Faibish, "Much ado about ChatGPT vs DeepSeek," *Computer*, vol. 58, no. 9, pp. 108-111, Sep. 2025, doi: 10.1109/MC.2025.3573422.
3. S. Faibish, "Gemini vs. ChatGPT and DeepSeek: Much ado about crawling," *Computer*, vol. 58, no. 10, pp. 98-101, Oct. 2025, doi: 10.1109/MC.2025.3581405.
4. D. Amodei. "On DeepSeek and export controls." darioamodei.com. Accessed: Jan. 10, 2025. [Online]. Available: https://darioamodei.com/on-deepseek-and-export-controls
5. "DeepSeek FAQ." Stratechery by Ben Thompson. Accessed: Feb. 22, 2025. [Online]. Available: https://stratechery.com/2025/deepseek-faq/
6. Z. B. Akhtar, "From bard to Gemini: An investigative exploration journey through Google's evolution in conversational AI and generative AI," *Comput. Artif. Intell.*, vol. 2, no. 1, 2024, Art. no. 1378, doi: 10.59400/cai.v2i1.1378.
7. K. Thompson and D. Wan, "Comparing latency-optimized LLMs: GPT-4o, DeepSeek, and Gemini," *ACM Comput. Surv.*, vol. 58, no. 1, pp. 1–38, Jan. 2025, doi: 10.1145/3650101.
8. A. Elbaz and H. Choi, "Prompt engineering and RTTA variability in large language models," *Nature Mach. Intell.*, vol. 7, pp. 45–55, Jan. 2025, doi: 10.1038/s42256-025-00601-x.
9. C. D'Souza and M. Rahman, "Fine-tuning vs. crawling: Data freshness in LLMs," in *Proc. NeurIPS Workshop Real-Time AI*, San Diego, CA, USA, 2024, pp. 109–120.
10. L. Jiang, R. Behnke, and T. Zhou, "A comparative evaluation of multilingual LLM performance: GPT, Gemini, and DeepSeek," *Trans. ACL*, vol. 13, pp. 221–239, Feb, 2025.
11. "ChatGPT-4 technical report," OpenAI, San Francisco, CA, USA, Mar. 2024. [Online]. Available: https://cdn.openai.com/papers/gpt-4.pdf

SORIN FAIBISH is a technology consultant in Newton, MA 02461 USA. Contact him at sfaibish@comcast.net.

Digital Object Identifier 10.1109/MC.2025.3638054