RULERv2: From Basic Retrieval to Complex Reasoning, A Bottom-Up Benchmark for Long-Context Evaluation

Cheng-Ping Hsieh, Faisal Ladhak, Krishna Puvvada, Boris Ginsburg
NVIDIA
{chsieh, fladhak, kpuvvada, bginsburg}@nvidia.com

Abstract

Recent progress in long-context language models has spurred development of diverse benchmarks that often test multiple skills simultaneously, making it difficult to pinpoint specific reasons for model failures. To address this, we introduce RULERv2, a benchmark with systematic difficulty progression from basic synthetic retrieval to complex problem-solving across three domains: multi-key NIAH, multi-value NIAH, and multi-doc QA. We conduct a large-scale evaluation of leading models, including seven closed-source and 27 open-weight models. Our findings reveal a notable performance gap between the two. Critically, we demonstrate that all models, including those claiming million-token context windows, exhibit performance degradation with increasing length, highlighting an unresolved challenge. Our analysis shows that explicit decomposition into a retrieve-then-solve strategy outperforms implicit, single-step approaches, and chain-of-thought reasoning enables models to discover effective decomposition autonomously. Finally, we find that even top-performing open-weight models struggle with fundamental retrieval and copying tasks, leading to degraded performance on more complex problems.[1]

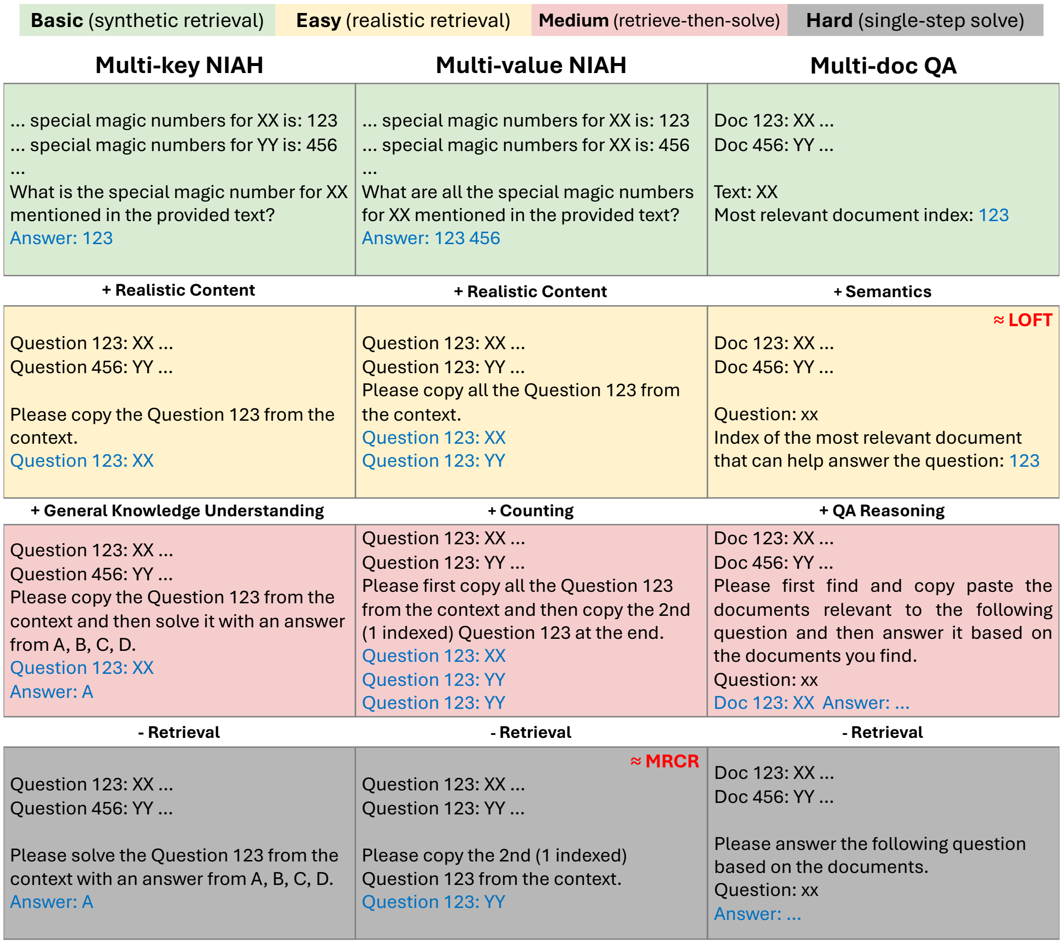
1 Introduction

Recent advancements in long-context language models have driven the creation of various evaluation benchmarks [1, 2, 3, 4]. These benchmarks test skills beyond simple literal matching [5], including semantic retrieval, summarization, question answering, in-context learning, and coding. However, they suffer from a critical limitation: by simultaneously testing multiple capabilities (retrieval, aggregation, reasoning, etc.) [6], they obscure whether failures stem from basic information access or higher-level reasoning, hindering targeted model improvement.

To address this gap, we introduce RULERv2, a benchmark designed with a systematic bottom-up approach that isolates fundamental long-context capabilities before testing their integration. Unlike existing benchmarks that focus on multi-task ability, RULERv2 increases task difficulty from nearly solved tasks, like retrieval, to show performance degradation and more precise diagnosis of model limitations. RULERv2 is organized into three task domains selected from RULERv1 [7] including (1) multi-key NIAH: from retrieving a single needle to solving a single problem among concatenated questions [8]. (2) multi-value NIAH: from retrieving multiple, shared-key needles to counting and copying specific indexed problems [9]. (3) multi-doc QA: from literal content matching to question answering requiring semantic retrieval and QA reasoning [10]. Each domain progresses through four difficulty levels: basic (synthetic retrieval), easy (realistic retrieval), medium (retrieve-then-solve), and hard (single-step solve) (see Figure 1).

[1] We release our code at `https://anonymous.4open.science/r/RULERv2`

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Evaluating the Evolving LLM Lifecycle: Benchmarks, Emergent Abilities, and Scaling.



Basic (synthetic retrieval) **Easy** (realistic retrieval) **Medium** (retrieve-then-solve) **Hard** (single-step solve)

Multi-key NIAH	Multi-value NIAH	Multi-doc QA
... special magic numbers for XX is: 123 ... special magic numbers for YY is: 456 ... What is the special magic number for XX mentioned in the provided text? Answer: 123	... special magic numbers for XX is: 123 ... special magic numbers for XX is: 456 ... What are all the special magic numbers for XX mentioned in the provided text? Answer: 123 456	Doc 123: XX ... Doc 456: YY ... Text: XX Most relevant document index: 123
+ Realistic Content	+ Realistic Content	+ Semantics
Question 123: XX ... Question 456: YY ... Please copy the Question 123 from the context. Question 123: XX	Question 123: XX ... Question 123: YY ... Please copy all the Question 123 from the context. Question 123: XX Question 123: YY	≈ LOFT Doc 123: XX ... Doc 456: YY ... Question: xx Index of the most relevant document that can help answer the question: 123
+ General Knowledge Understanding	+ Counting	+ QA Reasoning
Question 123: XX ... Question 456: YY ... Please copy the Question 123 from the context and then solve it with an answer from A, B, C, D. Question 123: XX Answer: A	Question 123: XX ... Question 123: YY ... Please first copy all the Question 123 from the context and then copy the 2nd (1 indexed) Question 123 at the end. Question 123: XX Question 123: YY Question 123: YY	Doc 123: XX ... Doc 456: YY ... Please first find and copy paste the documents relevant to the following question and then answer it based on the documents you find. Question: xx Doc 123: XX Answer: ...
- Retrieval	- Retrieval	- Retrieval
Question 123: XX ... Question 456: YY ... Please solve the Question 123 from the context with an answer from A, B, C, D. Answer: A	≈ MRCR Question 123: XX ... Question 123: YY ... Please copy the 2nd (1 indexed) Question 123 from the context. Question 123: YY	Doc 123: XX ... Doc 456: YY ... Please answer the following question based on the documents. Question: xx Answer: ...

Figure 1: RULERV2 has total 12 tasks among three task domains (multi-key/multi-value NIAH, multi-doc QA) with four task difficulties from basic to hard. See Appendix A for the full details.

We conduct a large-scale evaluation of leading long-context models, including Gemini 2.5 [11], GPT 5 [12], GPT 4.1 [13], o3 [14], Claude Sonnet 4 [15], Grok 4 [16], and 24 open-weight models ranging from 8B to over 100B parameters. Our evaluation reveals that all models, including those claiming million-token context windows [17, 18, 19], exhibit performance degradation as context length increases, challenging current claims of solved long-context understanding. Additionally, we find a notable performance gap between closed-source and open-weight models, with top open-weight models being mixture-of-experts (MoE) transformers in large sizes without hybrid architectures.

Our analysis yields several key insights. First, decomposing complex problems into a retrieve-then-solve strategy significantly outperforms direct single-step solving. Chain-of-thought enables models to discover effective decomposition autonomously. Second, while more few-shot demonstrations benefit larger models, majority voting provides negligible gains, though the maximum scores improve with more generations. Finally, top-performing open-weight models struggle with basic synthetic retrieval tasks when needle length or quantity increases, highlighting persistent copying issues that propagate to more complex problems. Therefore, building truly reliable long-context models requires focusing fundamental tasks before tackling more difficult problems.

2 Experiments

Data. We use MMLU [20] for both multi-key and multi-value NIAH and HotPotQA [21] for multi-doc QA. MMLU uses the 5-shot setting while multi-value NIAH uses four needles. We generate 100 samples for each context length: 8k, 16k, 32k, 64k, and 110k tokens (approximates 128k performance to reserve tokens for reasoning and output). While most tasks have exact-match ground truths, we

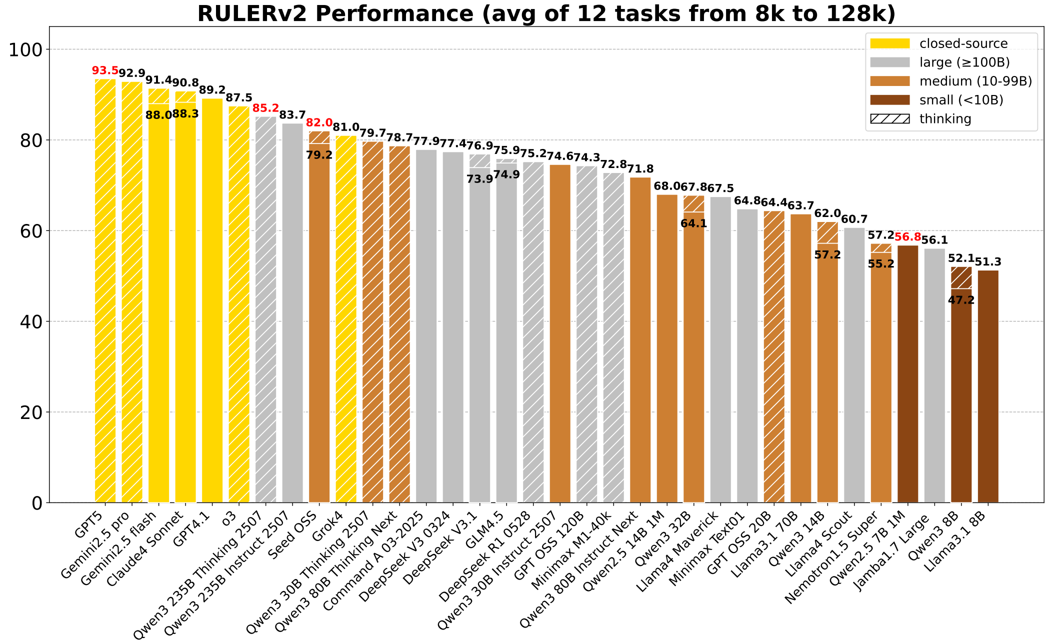


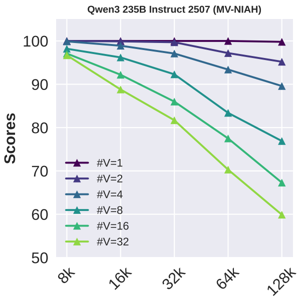
Figure 2: Performance of selected models on RULERV2. Scores are averaged across 12 tasks and five lengths ranging from 8k to 128k. Results with thinking are shown with stripes. The top-performing score for each model size is highlighted in red.

also account for partially correct answers. Our final score combines recall-based accuracy and word error rate (WER): $\max(\text{Recall}, 1 - \text{WER})$. Full inference setup is in Appendix B.

Models. We evaluate seven closed-source and 27 open-weight models, ranging from 7B to 671B parameters across dense transformers, hybrid transformers, and Mixture-of-Experts (MoE) architectures. Models span general instruction-following and enhanced reasoning capabilities. We use greedy decoding for instruct models and recommended sampling parameters for reasoning models, with 16k token output limits for Chain-of-Thought reasoning. Full specifications are in Appendix G.

Results Figure 2 shows aggregated RULERV2 performance. Except `Grok4`, closed-source models outperform open-weight models. Within model families (Qwen3), larger models consistently outperform smaller ones. Top-performing open-weight models are primarily MoE transformers, while hybrid architectures like Llama4, Minimax, and Jamba underperform expectations, trailing smaller dense transformers. Reasoning models show $3 - 7\%$ improvement when thinking is enabled. However, despite claims of context lengths beyond 128k, no model maintains stable performance as context expands (Figure 4). Larger models maintain higher absolute performance but show similar relative degradation. `Gemini2.5 flash` and `GPT4.1`, competitive at shorter lengths, suffer 15% degradation at 1M tokens (details in Appendix C).

3 Analysis



Number of needles. Our default multi-value NIAH uses four needles, but we analyze varying configurations. The figure on the right shows increasing needle count leads to performance degradation for the top-performing open-weight model, as it requires attending to multiple scattered locations. This indicates current models cannot reliably retrieve all relevant information, likely due to attention mechanism limitations. For other NIAH analysis, see Appendix D.1.

Task difficulties from basic to hard. Figure 3 (top left) shows all models achieve near-perfect scores on basic and easy multi-key NIAH, but performance drops at medium and hard levels relative to

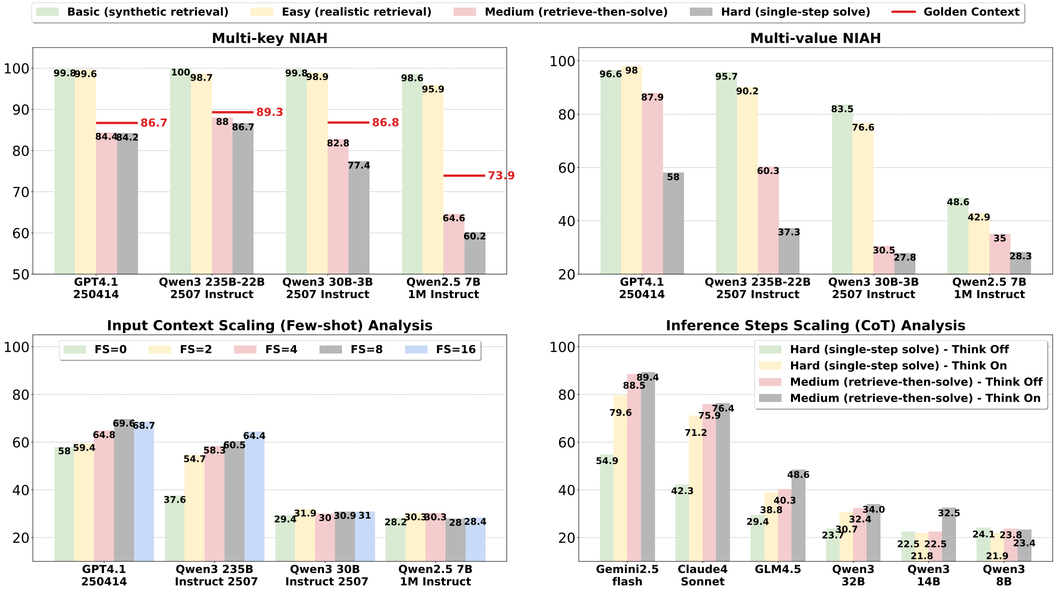


Figure 3: (**Top**) We analyze four task difficulties in two task domains, using four instruct models from each model sizes. Golden context is the score without any distractors. (**Bottom**) Results of test-time compute scaling including input context scaling and inference steps scaling. All the scores are evaluated on the hard level of multi-value NIAH task.

golden context baseline (short-context MMLU). For multi-value NIAH (Figure 3, top right), `GPT4.1` struggles with joint retrieval and counting (hard level), achieving only 58% accuracy. This explains why MRCR [9] is difficult: failure can stem from either retrieval or counting. Both analyses show retrieve-then-solve (medium level) consistently outperforms single-step solve (hard level), suggesting that these tasks can be effectively decomposed and accurate retrieval is a prerequisite for reliable downstream skill usage within larger contexts. Additional results in Appendix D.2.

Scaling test-time compute. We scale input via few-shot demonstrations from 0 to 16 (Figure 3, bottom left). Performance improves for larger models while smaller models show no improvement over zero-shot performance, suggesting a capacity threshold where models need sufficient parameters to learn complex retrieval and counting patterns from in-context examples. We evaluate reasoning models across medium and hard difficulty levels (Figure 3, bottom right). Results show hard task performance with thinking approaches medium task performance without thinking, suggesting reasoning enables autonomous task decomposition. For majority voting analysis see Appendix D.3.

4 Conclusion

We introduced RULERV2, a systematic bottom-up benchmark that progressively increases task difficulty from basic synthetic retrieval to complex problem-solving across three key domains. Through comprehensive evaluation of 34 long-context models, we revealed critical limitations that challenge existing claims of solved long-context understanding. All models, including those claiming million-token context windows, exhibit performance degradations as task difficulty and context length increase. Most importantly, even top-performing open-weight models struggle with fundamental retrieval and copying tasks which are prerequisite skills for complex problem-solving. This finding suggests that reliable long-context AI requires a foundation-first approach, where mastering basic information access precedes complex multi-step reasoning. Our analysis reveals that explicit task decomposition through the retrieve-then-solve strategy consistently outperforms single-step approaches, and chain-of-thought reasoning enables models to autonomously discover effective problem-solving strategies. RULERV2 addresses a critical evaluation gap by systematically isolating fundamental capabilities, providing a rigorous framework for diagnosing model limitations and measuring progress on the core skills that underpin reliable long-context understanding.

References

[1] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*, 2023.

[2] Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, et al. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*, 2024.

[3] Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. Helmet: How to evaluate long-context language models effectively and thoroughly. *arXiv preprint arXiv:2410.02694*, 2024.

[4] Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. ∞bench: Extending long context evaluation beyond 100k tokens. *arXiv:2402.13718*, 2024.

[5] Gregory Kamradt. Needle In A Haystack - pressure testing LLMs. *Github*, 2023.

[6] Jiaheng Liu, Dawei Zhu, Zhiqi Bai, Yancheng He, Huanxuan Liao, Haoran Que, Zekun Wang, Chenchen Zhang, Ge Zhang, Jiebin Zhang, et al. A comprehensive survey on long context language modeling. *arXiv preprint arXiv:2503.17407*, 2025.

[7] Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

[8] Xiang Liu, Peijie Dong, Xuming Hu, and Xiaowen Chu. Longgenbench: Long-context generation benchmark. *arXiv preprint arXiv:2410.04199*, 2024.

[9] Kiran Vodrahalli, Santiago Ontanon, Nilesh Tripuraneni, Kelvin Xu, Sanil Jain, Rakesh Shivanna, Jeffrey Hui, Nishanth Dikkala, Mehran Kazemi, Bahare Fatemi, et al. Michelangelo: Long context evaluations beyond haystacks via latent structure queries. *arXiv preprint arXiv:2409.12640*, 2024.

[10] Jinhyuk Lee, Anthony Chen, Zhuyun Dai, Dheeru Dua, Devendra Singh Sachan, Michael Boratko, Yi Luan, Séb Arnold, Vincent Perot, Siddharth Dalmia, et al. Loft: Scalable and more realistic long-context evaluation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6698–6723, 2025.

[11] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[12] OpenAI. Introducing gpt-5, 2025.

[13] OpenAI. Introducing gpt-4.1 in the api, 2025.

[14] OpenAI. Introducing openai o3 and o4-mini, 2025.

[15] Anthropic. Introducing claude 4, 2025.

[16] XAI. Grok 4, 2025.

[17] Meta. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation, 2025.

[18] Aonian Li, Bangwei Gong, Bo Yang, Boji Shan, Chang Liu, Cheng Zhu, Chunhao Zhang, Congchao Guo, Da Chen, Dong Li, et al. Minimax-01: Scaling foundation models with lightning attention. *arXiv preprint arXiv:2501.08313*, 2025.

[19] An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, et al. Qwen2. 5-1m technical report. *arXiv preprint arXiv:2501.15383*, 2025.

[20] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[21] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

[22] Samy Jelassi, David Brandfonbrener, Sham M Kakade, and Eran Malach. Repeat after me: Transformers are better than state space models at copying. *arXiv preprint arXiv:2402.01032*, 2024.

[23] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*, 2023.

[24] Woosuk Kwon et al. Efficient memory management for large language model serving with paged attention. In *Proc. of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[25] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Livia Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. Sglang: Efficient execution of structured language model programs. *Advances in neural information processing systems*, 37:62557–62583, 2024.

[26] Chenxin An, Fei Huang, Jun Zhang, Shansan Gong, Xipeng Qiu, Chang Zhou, and Lingpeng Kong. Training-free long-context scaling of large language models. *arXiv preprint arXiv:2402.17463*, 2024.

[27] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023.

[28] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[29] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[30] 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.

[31] Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Trung Bui, Ryan A Rossi, Seunghyun Yoon, and Hinrich Schütze. Nolima: Long-context evaluation beyond literal matching. *arXiv preprint arXiv:2502.05167*, 2025.

[32] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[33] Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. Zeroscrolls: A zero-shot benchmark for long text understanding. *arXiv preprint arXiv:2305.14196*, 2023.

[34] Chenxin An, Shansan Gong, Ming Zhong, Xingjian Zhao, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. L-eval: Instituting standardized evaluation for long context language models. *arXiv preprint arXiv:2307.11088*, 2023.

[35] Zican Dong, Tianyi Tang, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. Bamboo: A comprehensive benchmark for evaluating long text modeling capacities of large language models. *arXiv preprint arXiv:2309.13345*, 2023.

[36] Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. Loogle: Can long-context language models understand long contexts? *arXiv preprint arXiv:2311.04939*, 2023.

[37] Yury Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack. *Advances in Neural Information Processing Systems*, 37:106519–106554, 2024.

[38] Yang Zhou, Hongyi Liu, Zhuoming Chen, Yuandong Tian, and Beidi Chen. Gsm-infinite: How do your llms behave over infinitely increasing context length and reasoning complexity? *arXiv preprint arXiv:2502.05252*, 2025.

[39] Yifei Yu, Qian-Wen Zhang, Lingfeng Qiao, Di Yin, Fang Li, Jie Wang, Zengxi Chen, Suncong Zheng, Xiaolong Liang, and Xing Sun. Sequential-niah: A needle-in-a-haystack benchmark for extracting sequential needles from long contexts. *arXiv preprint arXiv:2504.04713*, 2025.

[40] Jonathan Roberts, Kai Han, and Samuel Albanie. Needle threading: Can llms follow threads through near-million-scale haystacks? *arXiv preprint arXiv:2411.05000*, 2024.

[41] Omer Goldman, Alon Jacovi, Aviv Slobodkin, Aviya Maimon, Ido Dagan, and Reut Tsarfaty. Is it really long context if all you need is retrieval? towards genuinely difficult long context nlp. *arXiv preprint arXiv:2407.00402*, 2024.

[42] Marzena Karpinska, Katherine Thai, Kyle Lo, Tanya Goyal, and Mohit Iyyer. One thousand and one pairs: A" novel" challenge for long-context language models. *arXiv preprint arXiv:2406.16264*, 2024.

[43] Fiction.liveBench. Fiction.livebench august 21 2025. 2025.

[44] Tianyang Liu, Canwen Xu, and Julian McAuley. Repobench: Benchmarking repository-level code auto-completion systems. *arXiv preprint arXiv:2306.03091*, 2023.

[45] Egor Bogomolov, Aleksandra Eliseeva, Timur Galimzyanov, Evgeniy Glukhov, Anton Shapkin, Maria Tigina, Yaroslav Golubev, Alexander Kovrigin, Arie Van Deursen, Maliheh Izadi, et al. Long code arena: a set of benchmarks for long-context code models. *arXiv preprint arXiv:2406.11612*, 2024.

[46] Amanda Bertsch, Maor Ivgi, Emily Xiao, Uri Alon, Jonathan Berant, Matthew R Gormley, and Graham Neubig. In-context learning with long-context models: An in-depth exploration. *arXiv preprint arXiv:2405.00200*, 2024.

[47] Kaijian Zou, Muhammad Khalifa, and Lu Wang. On many-shot in-context learning for long-context evaluation. *arXiv preprint arXiv:2411.07130*, 2024.

[48] Qwen. Qwen3 technical report, 2025.

[49] DeepSeek-AI. Deepseek-v3 technical report, 2024.

[50] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

[51] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

[52] Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

[53] Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, et al. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025.

[54] ByteDanceSeed. Seed-oss open-source models. `https://github.com/ByteDance-Seed/seed-oss`, 2025.

[55] Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*, 2025.

[56] Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, Omri Puny, Ido Galil, Zach Moshe, Tomer Ronen, Najeeb Nabwani, et al. Llama-nemotron: Efficient reasoning models. *arXiv preprint arXiv:2505.00949*, 2025.

[57] Barak Lenz, Alan Arazi, Amir Bergman, Avshalom Manevich, Barak Peleg, Ben Aviram, Chen Almagor, Clara Fridman, Dan Padnos, et al. Jamba-1.5: Hybrid transformer-mamba models at scale. *arXiv preprint arXiv:2408.12570*, 2024.

[58] Team Cohere, Arash Ahmadian, Marwan Ahmed, Jay Alammar, Milad Alizadeh, Yazeed Alnumay, Sophia Althammer, Arkady Arkhangorodsky, Viraat Aryabumi, Dennis Aumiller, et al. Command a: An enterprise-ready large language model. *arXiv preprint arXiv:2504.00698*, 2025.

A Benchmark Details

The RULERV2 benchmark comprises three task domains: multi-key NIAH, multi-value NIAH, and multi-doc QA. Each category features four levels of difficulty designed to test long-context capabilities: basic (synthetic retrieval), easy (realistic retrieval), medium (retrieve-then-solve), and hard (single-step solve). This progression enables systematic analysis of model capabilities. If a model fails at the basic/easy level, the issue is fundamental retrieval. If it succeeds at the medium level but fails at hard, the problem lies in implicit task decomposition rather than underlying skill. All evaluation examples are extensible to arbitrary context lengths and allow for flexible substitution of the base tasks. See Appendix H for the full task templates.

A.1 Multi-key Needle-in-a-Haystack (Multi-key NIAH)

Based on the definition from [7], this task requires models to retrieve a single "needle" (a target piece of information) from a context filled with similar distractor needles. Prior work has studied variants of this task, such as phonebook lookups [22] and JSON key-value retrieval [23]. Although finding specific information among highly similar content is challenging, some state-of-the-art long-context models can already achieve near perfect scores on such tasks. Therefore, we view the original MK-NIAH task as a foundational skill and increase the difficulty by requiring the model to not only retrieve but also solve a problem from a set of concatenated questions.

- **Basic:** The context consists of key-value pairs, where a word (the key) maps to a number (the value). Given a query word, the model must retrieve the corresponding number. This tests pure synthetic retrieval capability.
- **Easy:** We use a question index as the needle key and a question sampled from a short-context task as the needle value. Given an index, the model must retrieve and copy the corresponding question. This tests retrieval with realistic content.
- **Medium:** Beyond copying the question, this level tests if the model can also solve it. The prompt instructs the model to first retrieve and copy the question, and then provide its answer. This tests a combination of retrieval and problem-solving.
- **Hard:** Without an explicit instruction to retrieve and copy, this level tests if the model can directly solve one of the concatenated questions. This tests implicit task decomposition, i.e. whether models can autonomously break down the complex task into retrieval and solving steps. The performance upper bound is the score on the golden question presented alone.

A.2 Multi-value needle-in-a-Haystack (Multi-value NIAH)

Unlike multi-key NIAH, this task requires the model to find all needles that share the same key. The goal is to evaluate the ability to perform comprehensive retrieval of non-unique information. We increase the difficulty by introducing a counting component to find a needle at a specific ordinal position, similar to the Multi-Round Co-reference Resolution (MRCR) task [9]. Both tasks require identifying all needles with the same key scattered in the context and then using their order to determine the correct answer.

- **Basic:** The context contains multiple needles sharing the same key (a word). Given the query word, the model must retrieve all associated values (numbers). This tests the ability to perform multiple selective recalls.
- **Easy:** The needles are changed to indices mapping to realistic questions. Given an index, the model must retrieve and copy all questions associated with that same index, introducing realistic content to comprehensive retrieval.
- **Medium:** After retrieving all relevant questions, the model must identify a specific question by ordinal position. This adds a counting component to test whether models can track order relationships among retrieved information.
- **Hard:** Without explicit retrieval instructions, the model must directly find the question at a specific ordinal position for the given key. This task simplifies MRCR by removing confounding factors such as multi-turn conversational history and the requirement to prepend special strings, while preserving the core challenge of joint retrieval and counting.

A.3 Multi-document question answering (multi-doc QA)

This task tests capabilities directly relevant to Retrieval-Augmented Generation (RAG), and has been used in various forms in prior benchmarks [1, 3, 10]. In RULERV2, we progress from literal, exact-match retrieval to more complex sematic retrieval and question answering.

- **Basic:** We test the fundamental skill of document retrieval. The model is provided with the full content of a document and must retrieve its correct index. This isolates pure document identification capability.
- **Easy:** This task requires the model to perform semantic retrieval by identifying documents that contain the answer to a given question. This test a model the ability to understand conceptual relationships between a query and document content.
- **Medium:** This level tests QA reasoning after explicit retrieval. The model must first copy and paste the relevant documents and then use them to answer the question.
- **Hard:** Without explicit retrieval instructions, the model is tested on its ability to perform end-to-end QA by directly answering the question based on the full context of concatenated documents, requiring both semantic understanding and reasoning integration.

B Additional Experimental Setup

To efficiently speed up inference, we evaluated all open-weight models using vLLM [24] and SGLang [25]. We ran inference on a maximum of 16 NVIDIA H100 GPUs, depending on the hardware requirements recommended by the source. All tasks were completed in under four hours, with the exception of the 1m token experiment.

C Additional Results

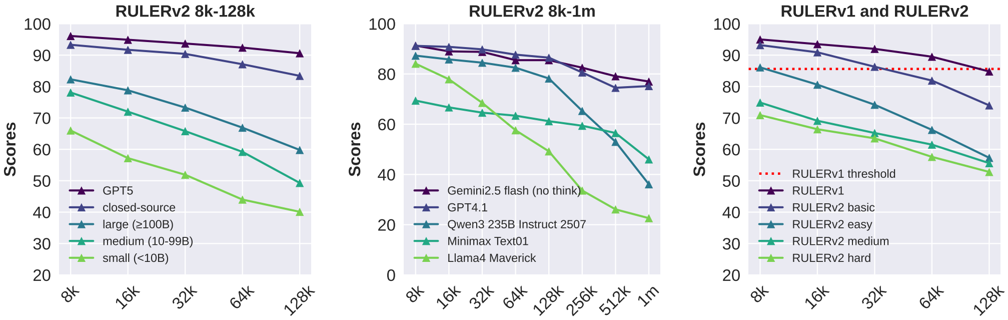


Figure 4: (**Left**): Comparison of different model sizes from lengths 8k to 128k. (**Middle**): Comparison of models claiming 1m context length. (**Right**): Comparison of RULERV1 and RULERV2.

Performance Degradation with Context Length. Increasing context length reveals a universal challenge across all models. Figure 4 (left) plots performance from 8k to 128k tokens, demonstrating that no model maintains stable performance as context expands, even though most of these models claim context lengths well beyond 128k. Even `GPT5`, which achieves the highest absolute scores, exhibits a significant performance degradation when going from 8k to 128k context length, despite claims of 400k context capability. The degradation patterns vary by model size and architecture: larger models maintain higher absolute performance across all context lengths but show similar relative decline rates, while open-weight models exhibit more severe degradation compared to the closed-source counterparts. These results suggest that while model size scaling improves baseline performance, it does not necessarily address the core challenge of maintaining performance as context increases.

Performance up to 1M Context Length. Recent claims of million-token context windows require further examination. Figure 4 (middle) extends our evaluation to 1m tokens for capable models, revealing substantial limitations. Both `Gemini 2.5 flash` and `GPT4.1`, which perform competitively at shorter lengths, suffer a significant degradation (approximately 15%) as the context extends to 1m tokens. Open-weight models face even steeper challenges. `Qwen3`, which is competitive with closed-source models under 128k, experiences sharp degradation beyond 256k tokens, despite employing length extrapolation techniques like dual chunk attention [26] and attention temperature scaling [27]. This suggests current extrapolation methods remain insufficient for reliable ultra-long context processing. Hybrid architectures show mixed results at extended lengths. `Llama4` exhibits a sharp degradation across all lengths, whereas `Minimax` maintains stable performance as context increases but struggles to achieve high scores even at short lengths. These results suggest that while hybrid architectures are computationally efficient, they have not yet matched the performance as full-attention transformers for long-context processing.

Comparison between RULERV1 and RULERV2. Comparison with RULERV1 demonstrates the evolving landscape of long-context capabilities. When RULERV1 was released, most models showed significant performance degradation at long contexts. Figure 4 (right) shows that current models achieve high, often saturated, scores on RULERV1, surpassing its established performance thresholds. This saturation indicates that many existing benchmarks may no longer effectively differentiate model capabilities or identify remaining challenges.

RULERV2 addresses these limitations by providing substantial headroom for improvement. Performance decreases systematically as both context length and task difficulty increase, creating a challenging evaluation space with clear improvement targets. The difficulty progression from basic retrieval to complex reasoning ensures that even models achieving perfect scores on simpler tasks encounter meaningful challenges at higher levels. This ability to reveal performance gaps where other benchmarks show saturation confirms the utility of RULERV2 for measuring progress on fundamental long-context skills.

D Additional Analysis

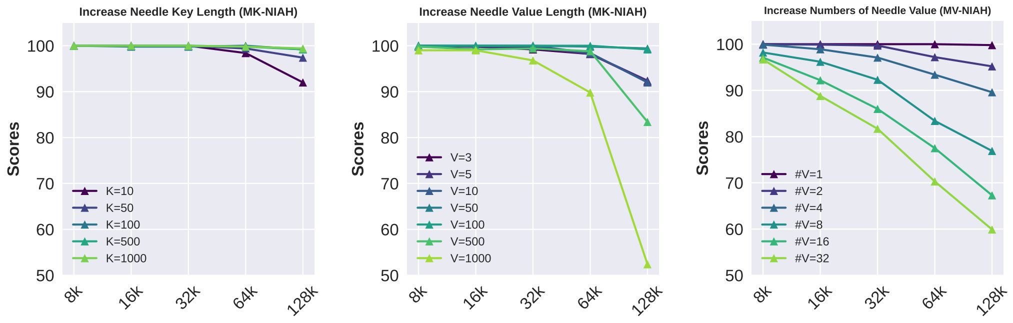


Figure 5: Needle-in-a-haystack variants by increasing needle key length (**left**) and needle value length (**middle**) in multi-keys NIAH as well as the numbers of needle value (**right**) in multi-values NIAH. We use model `Qwen3 235B-22B 2507 Instruct` for this analysis.

D.1 Needle-in-a-Haystack Variants

Following RULERV1, which proposed several NIAH variants by altering the type and quantity of needles and haystacks, we analyze the impact of varying the needle's key length, value length, and the number of values associated with a single key. The realistic retrieval (easy levels) of our multi-key and multi-value NIAH tasks can be viewed as a practical form of increasing the needle's value length.

Increase needle key length. To analyze the effect of key length, we use numbers with an increasing number of digits as the needle key, with results shown in Figure 5 (left). We observe performance degradation when using keys with 10 and 50 digits, but the model performance is stable for larger key lengths. This suggests that longer needle keys are actually easier for the model to locate, likely because they provide more distinctive patterns that reduce ambiguity and false matches with other subsequences in the context.

Increase needle value length. As shown in Figure 5 (middle), increasing the needle value length to 500 or 1000 digits leads to significant performance degradation. This occurs because the model struggles to accurately copy extremely long and contiguous sequences from the context. Interestingly, we also observe a performance drop for very short values (3, 5, and 10 digits). This may be because under a fixed context length budget, shorter needle values result in higher needle density, leading to substantially more distractors within the context. This increased distractor density makes the retrieval task more challenging despite the individual needles being shorter.

Increase numbers of needle value. Our default Multi-value NIAH task uses four needles, but other work has explored different configurations [11, 13]. In Figure 5 (right), we show that increasing the number of needles leads to progressive performance degradation, as it requires the model to attend to multiple scattered locations within the context. This analysis indicates that current models cannot reliably retrieve all relevant information, making this a key ongoing challenge. We hypothesize that the root cause may stem from fundamental limitations of the attention mechanism, a direction that requires further investigation.

D.2 Task Difficulties from Basic to Hard

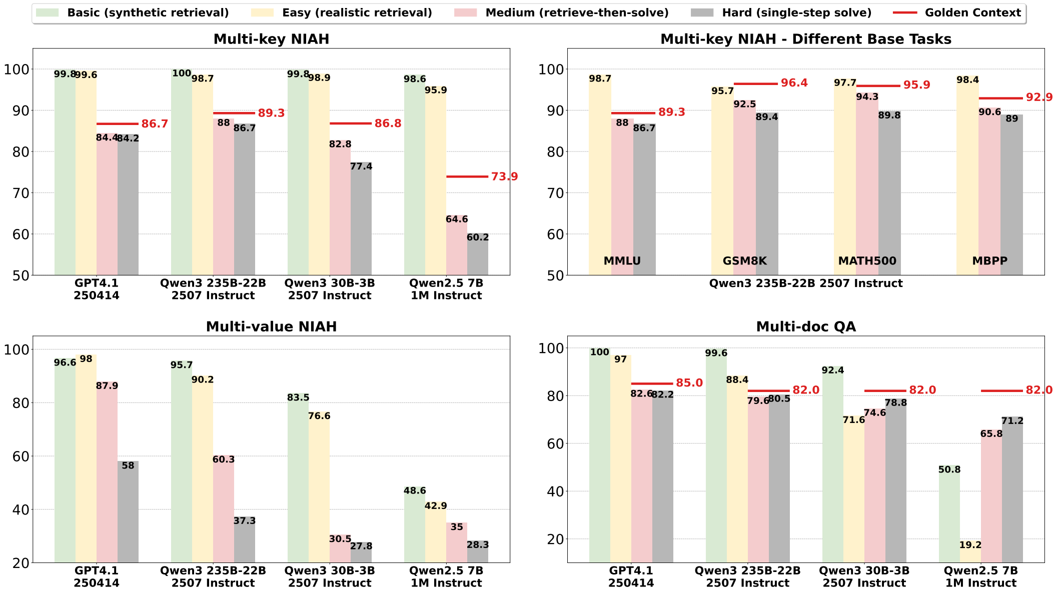


Figure 6: We analyze four task difficulties in three task domains, using four instruct models from each model sizes. Golden context is the score without any distractors.

RulerV2 comprises 12 tasks across three domains and four difficulty levels. We analyze the performance of four different model sizes across these difficulties, plotting the results in Figure 6.

Multi-key NIAH. In Figure 6 (top left), all models achieve near-perfect scores on the basic and easy levels, though 7B model exhibits a clear degradation from basic to easy. At the medium and hard levels, all models show a performance drop relative to baseline task performance (i.e., their short-context MMLU score). We observe a similar trend when substituting other base tasks like GSM8K [28], MATH500 [29], and MBPP [30], with the degradation being particularly severe for math-related benchmarks (top right). Notably, performance on the medium level (retrieve-then-solve) is consistently higher than on the hard level (single-step solve). This suggests that this task can be effectively decomposed, and good retrieval is a prerequisite for reliably solving a specific question embedded within a larger context.

Multi-value NIAH. As shown in Figure 6 (bottom left), open-source models perform poorly even the basic level, with the smaller 7B model achieving only a 48.6% accuracy. Even `GPT4.1`, which demonstrates near-perfect retrieval, struggles to jointly perform retrieval and counting in the hard setting, achieving only a 58% accuracy. This finding helps explain why MRCR [9] is difficult: failure can stem from either the retrieval or the counting skill. Decomposing the task into explicit retrieval and then counting under medium setting results in a substantial performance improvement for all models compared to the harder setting. Analysis of model outputs reveals that successful medium-level responses typically first enumerate all relevant questions before performing the ordinal selection, while hard-level failures often result from models attempting to count implicitly without explicit enumeration.

Multi-doc QA. The results in Figure 6 (bottom right) show that open-weight models cannot perfectly solve even the basic synthetic retrieval task. When semantic understanding is required at the easy level, all models degrade, likely due to failures in understanding latent associations between query and document beyond literal keyword matching [31]. At higher difficulty levels, all models fail to match their golden context performance. Counter-intuitively, for smaller models, hard level scores sometimes exceed medium-level scores. Manual inspection of outputs suggests that smaller models, even without explicit retrieval instructions, implicitly adopt a retrieve-then-answer strategy, by incorporating relevant document content into their responses, though they may paraphrase rather than exactly copy the source material, which can complicate exact-match [32] metrics.

D.3 Scaling Test-Time Compute

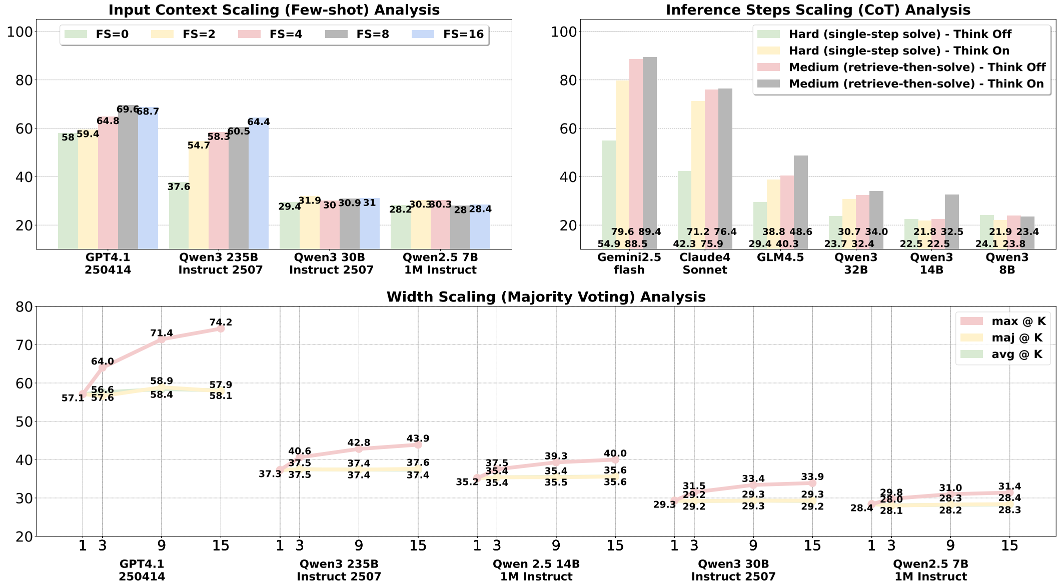


Figure 7: Results of test-time compute scaling including input context scaling (**top left**), inference steps scaling (**top right**), and width scaling (**bottom**). All the scores are averaged across lengths 8k to 128k and evaluated on the hard level of multi-values NIAH task.

We explore three test-time compute scaling methods: (1) scaling input context via few-shot demonstrations, (2) scaling inference steps via Chain-of-Thought reasoning, and (3) scaling width via majority voting. We use the most challenging task, multi-value NIAH (hard), for this analysis, with results in Figure 7.

Scaling Input Context. We scale the number of few-shot demonstrations from 0 to 16. As shown in Figure 7 (top left), performance improves for larger models, while smaller models show no improvement over their zero-shot performance. This suggests a capacity threshold: models require sufficient parameters to effectively learn complex retrieval and counting patterns from in-context examples.

Scaling Inference Steps. We evaluate models with reasoning capability across both medium and hard difficulty levels. Figure 7 (top right) confirms the benefit of generating a chain of thought before the final answer, with all except the smallest models showing improvements. Crucially, performance on the hard task with thinking approaches the performance on the medium task without thinking, suggesting that explicit reasoning allows the model to autonomously decompose complex tasks. Manual analysis of generated reasoning chains reveals that models consistently follow a retrieve-then-count pattern: first enumerating all relevant information, and then performing ordinal selection. This mirrors our benchmark decomposition (medium setting), demonstrating that reasoning-capable models can autonomously discover problem-solving strategies.

Scaling Width. We generate 1 to 15 parallel responses and analyze maximum, majority, and average scores in Figure 7 (bottom). Majority voting provides little improvements, as models consistently produce similar incorrect responses rather than diverse attempts. However, maximum scores across all generations steadily increases, especially for larger models. For instance, the performance gain from 1 to 15 generations is 6.6% for the largest open-weight model versus 3.0% for the smallest. This suggests that while the primary prediction is stable, larger model possess a broader solution spaces and occasionally generate correct responses which could be identified through more selective sampling strategies.

E Related Work

Existing long-context benchmarks typically aim for comprehensive evaluation by integrating a diverse set of use cases, including retrieval, question answering, summarization, in-context learning, and coding. While early versions were limited to shorter contexts [33, 34, 1, 35, 36] because of the rapid extension of model context lengths, others can evaluate models with context lengths from 128k to over a million tokens [4, 7, 3, 2, 10]. Alongside these multi-task benchmarks, several synthetic tests have been proposed to reliably verify long-context capabilities. The widely used "Needle-in-a-Haystack" (NIAH) test [5] assesses a model's ability to recall a fact from an extremely long document. Other examples include MRCR [9], which tests recalling conversations from a specific ordinal position; NoLiMa [31], which requires inferring latent associations; BABILong [37] and GSM-∞ [38], which focus on reasoning across facts in complex contexts; Sequential-NIAH [39], designed for extracting sequential information; NeedleThreading [40], evaluating the ability to follow threads of information. A common spirit among these tasks is their needs on retrieval, a fundamental capability for processing long contexts [41]. Therefore, we developed RULERV2 by starting with simple retrieval tests and progressively increasing task difficulty in a bottom-up manner. Our benchmark is designed to identify the limitations of current long-context models, highlighting the need for improving fundamental retrieval abilities before tackling more complex reasoning challenges.

F Limitations

This study has several limitations that we have considered and describe in detail below.

Lack of correlation with realistic long-context tasks. All of our tasks are designed to evaluate long-context skills synthetically. This is because there are no realistic long-context tasks that are easy to scale to millions of tokens and can be automatically evaluated without manual checking. While we emphasize our benchmark as a convenient check to verify long-context capabilities, we still need to test models in realistic settings that are closer to how they would be truly used. These settings would require multiple capabilities beyond retrieval, such as question answering from books [42, 43], processing code repository [44, 45], and many-shot in-context learning [46, 47].

Lack of a clear definition of fundamental skills. Our bottom-up benchmark is built to progressively increase in difficulty, starting with retrieval and moving on to problem-solving skills. However, we need a clearer definition of the fundamental skills required for other long-context tasks and how they relate to retrieval, such as aggregation and reasoning. Although this survey [6] has defined retrieval as the foundation, more studies are needed to break down difficult and complex long-context tasks into fundamental skills. This would help us better analyze failures and understand the limitations of current models.

G Models

We select in total 37 models (listed in Table 1) for evaluation including 7 closed-source and 27 open-weight models. In open-weight models, we have 12 large size (≥100B), nine medium size (10−99B), and three small size (<10B). Regarding architectures, we have nine dense transformer and four hybrid transformers with 15 using Mixture-of-Experts. All the models have claimed their context length more than 128k tokens. Some of them have built-in reasoning switch to turn on and off thinking mode.

Table 1: Summary of the selected models in RULERV2.

Model Name	Size	Claimed Length	Thinking	Huggingface / API
GPT5 [12]	-	400k	Y	gpt-5 (2025-08-07)
GPT4.1 [13]	-	1m	N	gpt-4.1 (2025-04-14)
o3 [14]	-	200k	Y	o3 (2025-04-16)
Gemini2.5 pro [11]	-	1m	Y	gemini-2.5-pro
Gemini2.5 flash [11]	-	1m	Y/N	gemini-2.5-flash
Claude4 Sonnet [15]	-	200k	Y/N	claude-sonnet-4-20250514
Grok4 [16]	-	256k	Y	grok-4-0709
Qwen3 235B Thinking 2507 [48]	235B-22B	1m	Y	Qwen/Qwen3-235B-A22B-Thinking-2507
Qwen3 235B Instruct 2507 [48]	235B-22B	1m	N	Qwen/Qwen3-235B-A22B-Instruct-2507
Qwen3 30B Thinking 2507 [48]	30B-3B	1m	Y	Qwen/Qwen3-30B-A3B-Thinking-2507
Qwen3 30B Instruct 2507 [48]	30B-3B	1m	N	Qwen/Qwen3-30B-A3B-Instruct-2507
Qwen3 80B Thinking Next [48]	80B-3B	1m	Y	Qwen/Qwen3-Next-80B-A3B-Thinking
Qwen3 80B Instruct Next [48]	80B-3B	1m	N	Qwen/Qwen3-Next-80B-A3B-Instruct
Qwen3 32B [48]	32.8B	128k	Y/N	Qwen/Qwen3-32B
Qwen3 14B [48]	14.8B	128k	Y/N	Qwen/Qwen3-14B
Qwen3 8B [48]	8.2B	128k	Y/N	Qwen/Qwen3-8B
Qwen2.5 14B 1M [19]	14.7B	1m	N	Qwen/Qwen2.5-14B-Instruct-1M
Qwen2.5 7B 1M [19]	7.6B	1m	N	Qwen/Qwen2.5-7B-Instruct-1M
DeepSeek V3.1 [49]	671B-37B	128k	Y/N	deepseek-ai/DeepSeek-V3.1
DeepSeek R1 0528 [50]	671B-37B	128k	Y	deepseek-ai/DeepSeek-R1-0528
DeepSeek V3 0324 [49]	671B-37B	128k	N	deepseek-ai/DeepSeek-V3-0324
Llama4 Maverick [17]	400B-17B	1m	N	meta-llama/Llama-4-Maverick-17B-128E-Instruct-FP8
Llama4 Scout [17]	109B-17B	10m	N	meta-llama/Llama-4-Scout-17B-16E-Instruct
Llama3.1 70B [51]	70B	128k	N	meta-llama/Llama-3.1-70B-Instruct
Llama3.1 8B [51]	8B	128k	N	meta-llama/Llama-3.1-8B-Instruct
GPT OSS 120B [52]	117B-5.1B	128k	Y	openai/gpt-oss-120b
GPT OSS 20B [52]	21B-3.6B	128k	Y	openai/gpt-oss-20b
MiniMax M1-40k [53]	456B-45.9B	1m	Y	MiniMaxAI/MiniMax-M1-40k
MiniMax Text01 [18]	456B-45.9B	1m	N	MiniMaxAI/MiniMax-Text-01
Seed OSS [54]	36B	512k	Y/N	ByteDance-Seed/Seed-OSS-36B-Instruct
GLM 4.5 [55]	355B-32B	128k	Y/N	zai-org/GLM-4.5
Nemotron1.5 Super [56]	49.9B	128k	Y/N	nvidia/Llama-3_3-Nemotron-Super-49B-v1_5
Jamba1.7 Large [57]	398B-94B	256k	N	ai21labs/AI21-Jamba-Large-1.7
Command A 03-2025 [58]	111B	256k	N	CohereLabs/c4ai-command-a-03-2025

H Task Templates

The detailed task templates we used are provided in Tables 2, 3, and 4. We used MMLU [20] as the base task for our multi-key and multi-value NIAH tasks and used HotPotQA [21] for the multi-doc QA task.

Table 2: Multi-key NIAH templates from basic to hard difficulties.

Multi-key NIAH (Basic) Synthetic Retrieval	**Prompt:** A special magic number is hidden within the following text. Make sure to memorize it. I will quiz you about the number afterwards. One of the special magic numbers for word-1 is: number-1. One of the special magic numbers for word-2 is: number-2. ...... One of the special magic numbers for word-x is: number-x. ...... One of the special magic numbers for word-n-1 is: number-n-1. One of the special magic numbers for word-n is: number-n. What is the special magic number for word-x mentioned in the provided text? The special magic number for word-x mentioned in the provided text is **Answer:** number-x
Multi-key NIAH (Easy) Realistic Retrieval	**Prompt:** Below are some questions. I will ask you to copy one of them. Please copy and paste the question you find. Question index-1: question-1. Question index-2: question-2. ...... Question index-x: question-x. ...... Question index-n-1: question-n-1. Question index-n: question-n. Please copy the Question index-x from the context. **Answer:** question-x
Multi-key NIAH (Medium) Retrieve-then -solve	**Prompt:** Below are some questions. I will ask you to solve one of them. Please solve the question you find and make sure to put the answer (and only answer) inside \boxed{ }. Question index-1: question-1. Question index-2: question-2. ...... Question index-x: question-x. ...... Question index-n-1: question-n-1. Question index-n: question-n. Here are some examples to help you understand the task: ($\times N$) Please copy the Question index from the context and then solve it with an answer from A, B, C, D. Question index: question. Solution: \boxed{answer}. Here is the actual task you need to solve: Please copy the Question index-x from the context and then solve it with an answer from A, B, C, D. **Answer:** \boxed{answer-x}
Multi-key NIAH (Hard) Single-step Solve	**Prompt:** Below are some questions. I will ask you to solve one of them. Please solve the question you find and make sure to put the answer (and only answer) inside \boxed{ }. Question index-1: question-1. Question index-2: question-2. ...... Question index-x: question-x. ...... Question index-n-1: question-n-1. Question index-n: question-n. Here are some examples to help you understand the task: ($\times N$) Please solve the Question index from the context with an answer from A, B, C, D. Solution: \boxed{answer}. Here is the actual task you need to solve: Please solve the Question index-x from the context with an answer from A, B, C, D. **Answer:** \boxed{answer-x}

Table 3: Multi-value NIAH templates from basic to hard difficulties.

Multi-value NIAH (Basic) Synthetic Retrieval	**Prompt:** Some special magic numbers are hidden within the following text. Make sure to memorize them. I will quiz you about the numbers afterwards. One of the special magic numbers for word-1 is: number-1. ...... One of the special magic numbers for word-x is: number-x1. One of the special magic numbers for word-x is: number-x2. One of the special magic numbers for word-x is: number-x3. One of the special magic numbers for word-x is: number-x4. ...... One of the special magic numbers for word-n is: number-n. What are all the special magic numbers for word-x mentioned in the provided text? The special magic numbers for word-x mentioned in the provided text are **Answer:** number-x1, number-x2, number-x3, number-x4
Multi-value NIAH (Easy) Realistic Retrieval	**Prompt:** Below are some questions. I will ask you to copy some of them. Please copy and paste the questions you find. Question index-1: question-1. Question index-2: question-2. ...... Question index-x: question-x1. Question index-x: question-x2. Question index-x: question-x3. Question index-x: question-x4. ...... Question index-n-1: question-n-1. Question index-n: question-n. Please copy the Question index-x from the context. **Answer:** question-x1, question-x2, question-x3, question-x4
Multi-value NIAH (Medium) Retrieve-then -solve	**Prompt:** Below are some questions. I will ask you to copy one of them. Please copy and paste the question you find. Question index-1: question-1. Question index-2: question-2. ...... Question index-x: question-x1. Question index-x: question-x2. Question index-x: question-x3. Question index-x: question-x4. ...... Question index-n-1: question-n-1. Question index-n: question-n. Please first copy all the Question index-x from the context and then copy the {order} (1 indexed) Question index-x at the end. **Answer:** question-x{order}
Multi-value NIAH (Hard) Single-step Solve	**Prompt:** Below are some questions. I will ask you to copy one of them. Please copy and paste the question you find. Question index-1: question-1. Question index-2: question-2. ...... Question index-x: question-x1. Question index-x: question-x2. Question index-x: question-x3. Question index-x: question-x4. ...... Question index-n-1: question-n-1. Question index-n: question-n. Please copy the {order} (1 indexed) Question index-x from the context. **Answer:** question-x{order}

Table 4: Multi-doc QA templates from basic to hard difficulties.

Multi-doc QA (Basic) Synthetic Retrieval	**Prompt:** Below are some documents. I will give you a text at the end. Please find the document index of the text. Only give me the index without any document contents. Document index-1: document-1. Document index-2: document-2. ...... Document index-x: document-x. ...... Document index-n-1: document-n-1. Document index-n: document-n. Text: document-x Most relevant document index: **Answer:** index-x
Multi-doc QA (Easy) Realistic Retrieval	**Prompt:** Below are some documents. I will give you a question at the end. Please find the index of the most relevant document that can help answer the question. Only give me the index without any document contents. Document index-1: document-1. Document index-2: document-2. ...... Document index-x: document-x. ...... Document index-n-1: document-n-1. Document index-n: document-n. Question: question-x Index of the most relevant document that can help answer the question: **Answer:** index-x
Multi-doc QA (Meidum) Retrieve-then -solve	**Prompt:** Below are some documents. I will ask you to answer a question based on the documents. Please answer the question. Document index-1: document-1. Document index-2: document-2. ...... Document index-x: document-x. ...... Document index-n-1: document-n-1. Document index-n: document-n. Please first find and copy paste the documents relevant to the following question and then answer it based on the documents you find. Question: question-x **Answer:** answer-x
Multi-doc QA (Hard) Single-step Solve	**Prompt:** Below are some documents. I will ask you to answer a question based on the documents. Please answer the question. Document index-1: document-1. Document index-2: document-2. ...... Document index-x: document-x. ...... Document index-n-1: document-n-1. Document index-n: document-n. Please answer the following question based on the documents. Question: question-x **Answer:** answer-x