💯-LongBench: Are *de facto* Long-Context Benchmarks Literally Evaluating Long-Context Ability?

Wang Yang[1], **Hongye Jin**[2], **Shaochen Zhong**[3], **Song Jiang**[4], **Qifan Wang**[5]
Vipin Chaudhary[1], **Xiaotian Han**[1]
[1]Case Western Reserve University [2]Texas A&M University [3]Rice University
[4]University of California, Los Angeles [5]Meta
{wxy320,vipin,xhan}@case.edu, jhy0410@tamu.edu, hz88@rice.edu
songjiang@ucla.edu, wqfcr@meta.com

Abstract

Long-context capability is considered one of the most important abilities of LLMs, as a truly long context-capable LLM shall enable its users to effortlessly process many originally exhausting tasks — e.g., digesting a long-form document to find answers v.s., directly asking an LLM about it. However, existing real-task-based long-context evaluation benchmarks have a few major shortcomings. For instance, some Needle-in-a-Haystack-like benchmarks are too synthetic, and therefore do not represent the real world usage of LLMs. While some real-task-based benchmarks like LongBench avoid this problem, such benchmarks are often formed in a way where each data sample has a fixed sequence length, which not only makes them solely suitable for models with a certain range of context windows, but also lacks a proxy to know at what length the model/method-of-interest would fail. Last, most benchmarks tend to not provide proper metrics to separate long-context performance from the model's baseline ability, so when conducting a cross-model/recipe comparison, such conflation makes the user unable to understand how exactly one model or recipe excels at the long-context task in relation to its baseline ability. To address these issues, we introduce a length-controllable, real-life reflective benchmark with a novel metric that disentangles baseline knowledge from long-context capabilities. Experiments demonstrate the superiority of our datasets in effectively evaluating LLMs. All assets are available at https://github.com/uservan/100-LongBench.git.

1 Introduction

The long-context capability has become one of the fundamental competencies (Gao et al., 2024; Liu et al., 2024b; Li et al., 2024; Agarwal et al., 2024) of contemporary large language models (LLMs) because it takes the average human critical time and effort to digest long-form context, making a long-context-capable LLM beyond desirable. To assess the long-context capabilities of LLMs, various evaluation benchmarks and metrics have been proposed, including LongBench (Bai et al., 2023), L-Eval (An et al., 2023), NIAH (Needle in the Haystack), RULER (Hsieh et al., 2024), Ada-LEval (Wang et al., 2024) and Loogle (Li et al., 2023a). However, these benchmarks often exhibit at least one of the following three shortcomings:

Table 1: Models' ranking on Ruler (Hsieh et al., 2024) with different metrics. **Base Ability** represents model's score within $4k$ context. **Old/Proposed Metric** represents the average score across various lengths using traditional metric/our proposed metric. $96.5_{(1)}$ indicates a score of 96.5 with a rank of 1. More details are in Table 5. Comparing the ranking of Old Metric and Proposed Metric reveals that the rankings of the old metrics are heavily influenced by the model's inherent abilities, which might not really reflect long-context ability.

Model (size,length)	Base Ability	Old Metric	Proposed Metric
Llama3.1 (70B, 128K)	$96.5_{(1)}$	$88.2_{(1)}$	$-8.6_{(2)}$
Yi (34B, 200K) (Young et al., 2024)	$93.3_{(2)}$	$86.3_{(2)}$	$-7.5_{(1)}$
Phi3-medium (14B, 128K)	$93.3_{(3)}$	$79.1_{(3)}$	$-15.1_{(4)}$
LWM (7B, 1M) (Liu et al., 2024a)	$82.3_{(4)}$	$70.8_{(4)}$	$-13.9_{(3)}$

(1) They rely on purely **synthetically-constructed content that is not real-life reflective**. Synthetic benchmarks such as NIAH or Passkey Retrieval often demand the retrieval of a source (e.g., a string of digits or a phrase) that bears no semantic or task relevance to the padding content (e.g., unrelated blog posts). This kind of highly artificial task does not properly reflect how LLMs are utilized in typical real-world settings, where information of similar nature is often joined together for a reader to understand and digest.

(2) They adopt a **fixed input length per data sample**, making them suitable only for certain LLMs with compatible context windows. This is a major problem because context windows have grown significantly, thanks to the development of context extension techniques and post-training

Findings of the Association for Computational Linguistics: ACL 2025, pages 17560–17576

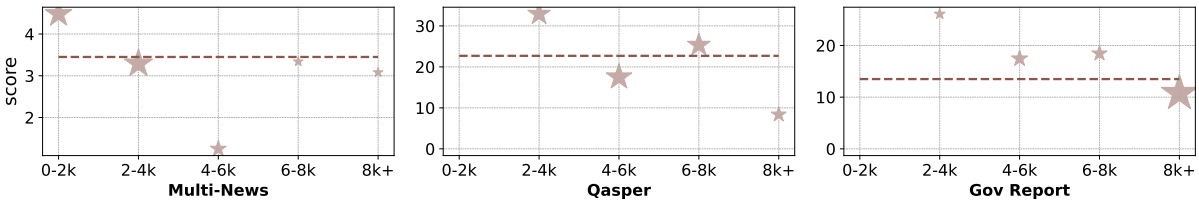

Figure 1: Illustration of LM-Infinite (Han et al., 2024), a long-context enhancement method's performances on three LongBench tasks. The colored dashed lines represent the average score of each model on the corresponding task. The size of the markers corresponds to the proportion of each text length within the entire dataset. The larger the marker, the higher the proportion. The results exhibit significant variation across tasks of different lengths within the same dataset. More results of other methods are in Appendix A.1.

recipes. With Llama 3.1 (Dubey et al., 2024) claiming a context window of 128k (in contrast to the 4k context of Llama 2), many once "long-context" datasets have already become outdated. It is therefore foreseeable that many evaluations we see today will no longer be reflective as time passes. Moreover, having different lengths per individual data sample makes the evaluation reading unintuitive in several ways: E.g., for model evaluation, it is hard to tell at what length it would fail or prevail, because we only get the aggregated reading upon questions of different lengths. For method evaluation, many constant-budget compression works — like StreamingLLM (Xiao et al., 2023a) and InfLLM (Xiao et al., 2024) — have an arbitrarily set constant budget that is applied to all inputs, ignoring the fact that this budget may exceed certain data samples. As a result, the reported "compressed performance" often turns into an unknown mixture of both compressed and uncompressed results, complicating the transparency of assessments.

(3) They do not address the **conflation between base ability and long-context capability**, as these benchmarks evaluate long-context capabilities solely based on long-context tasks without isolating the influence of a model's baseline abilities. Thus, some readings can be tricky to digest when factors cannot be perfectly ablated. For instance, suppose we have two different base models, each has undergone their own continuous pretraining recipes for context extension (e.g., Llama and Qwen), *which extension recipe is likely better?* Applying both recipes to the same base model for direct comparison is often impractical due to compute and dataset resource limitations. Naturally, one avenue of evaluation is to measure the long context performance relative to the short context performance for an educated understanding, but such kind of measurements is largely missing in most existing long-context benchmarks.

In this work, we attempt to alleviate such problems by providing a **new benchmark** involving a rich set of length-controllable real-life-reflective tasks — 💯-LongBench — and a **new evaluation metric** — LongScore — which leads to significant shifts in model rankings compared to traditional performance-based evaluations, as shown in Table 1. We first validate the reliability of the proposed 💯-LongBench and the effectiveness of LongScore. We then **comprehensively benchmark** various open-source models, providing **fresh insights** into long-context evaluation and offering a more accurate assessment that better reflects models' true abilities to handle extended contexts.

2 Motivation: why do we need to refine long-context benchmarks?

Performance variance across task lengths Evidenced by Figure 1, the performance of LM-Infinite exhibits significant variation across tasks of different lengths within the same dataset. Many long-context datasets have uneven length distributions, introducing biases in evaluating a model's long-context capability. To validate this hypothesis, we train models using five different long-context enhancement methods and evaluate their performances across varying lengths on the LongBench dataset. From Figure 1, we observe the following: (1) Performance Variation: All five models demonstrate performance differences across different text lengths within the same dataset. (2) Alignment with Dominant Lengths: A model's average performance aligns closely with its performance on the length range with the highest proportion of samples. For instance, on Multi-News dataset, each model's average performance is close to its performance on samples within the 0–4k length range, which represents the largest share of the dataset. These findings highlight the need for length-aware evaluations of long-context capabilities. A more robust approach

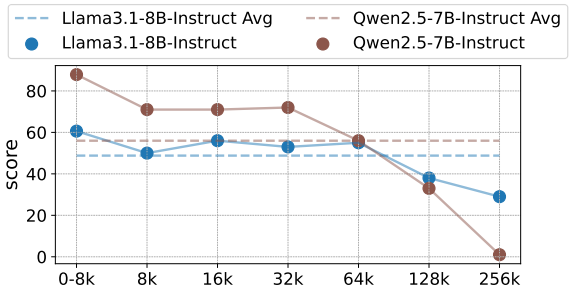

Figure 2: Comparison between LLaMA 3.1-8B-Instruct and Qwen 2.5-7B-Instruct on the Counting Star task across varying text lengths. The dashed line represents the average score across all context lengths. LLaMA 3.1-8B-Instruct performs worse than Qwen 2.5-7B-Instruct on short texts but excels on extremely long texts, indicating its superior long-context extension capability.

involves testing model performance on N samples across diverse lengths to obtain a comprehensive assessment of its long-context capability. More results of other methods are in Appendix A.1.

Ineffectiveness of current metrics for evaluating long-context capability Evidenced by Figure 2, existing long-context metrics primarily rely on the average performance across the benchmark. However, this approach can be misleading as it conflates the model's inherent task-specific ability with its pure long-context capability. As illustrated in Figure 2, LLaMA 3.1-8B-Instruct performs worse than Qwen 2.5-7B-Instruct on short texts but excels on extremely long texts, such as $128k$ and $255k$, indicating its superior long-context extension capability. In this task, the average performance suggests that Qwen 2.5-7B-Instruct is the better model. But a closer inspection reveals that LLaMA 3.1-8B-Instruct has a distinct advantage in handling extremely long texts, despite its weaker performance on shorter inputs. This discrepancy underscores the need to separate a model's base ability (on short texts) from its long-context capability. To address this, we propose a novel metric that accurately captures a model's true capability to handle long contexts from Base Ability.

3 How to truly evaluate Language Models' long-context capability?

To address the two identified problems, we 1) construct a length-controllable long-context benchmark to reduce performance variance across task lengths, and 2) introduce LongScore, a new metric designed to accurately evaluate long-context capabilities by disentangling the model's baseline abilities. In detail, we restructure the long-context datasets, based on LongBench, L-EVAL, and other benchmarks. We then design a new pipeline to generate controllable-length long contexts by combining different articles. Additionally, we introduce a filtering mechanism in QA-related tasks to mitigate prior knowledge. Subsequently, We propose a new metric to isolate a model's long-text capability from Base Ability (performance on short texts).

3.1 Construct a new long-context benchmark

We categorize tasks into four types, each consisting of two tasks with different levels of difficulty, resulting in a total of eight tasks. The types and their corresponding tasks are: ***Key Retrieval (including KV Retrieval and Counting Stars), Information Retrieval (including Passage Retrieval and Passage Count) , Information Comprehension (including Single-doc QA and Multi-doc QA) and Information Summarization (including Single-doc Sum and Multi-doc Sum)*** . Table 2 provides details for each task, including: Real Context Sources(the original context of the question used in the task), Noisy Context Sources(the source of additional context that may contain irrelevant or distracting information) and Evaluation Metric(the metric used to assess model performance for each task). All of these datasets are from other benchmarks like LongBench, etc. Detailed information on context construction, question setup, and evaluation metrics, are in Appendix A.2.

How to generate a controllable-length context? In 💯-LongBench, the context for each task is controllable, such as generating a context of approximately $128k$ tokens. To achieve this, we first randomly select one article from Real Context Sources as the ground truth article. Then, we randomly sample a number of articles from Noisy Context Sources as distractor articles. These distractor articles are combined with the ground truth article to construct the whole context, ensuring that the total context length is close to but less than $128k$. Finally, the order of all articles is shuffled to create the context. Figure 3 illustrates the data generation process for Single-Doc QA task, showing how questions, answers, and contexts are prepared.

QA Filtering Mechanism. For Multi-Doc QA and Single-Doc QA tasks, we introduce a filtering mechanism to eliminate the influence of the model's inherent prior knowledge. When evaluating a model's long-context capabilities, prior

Table 2: Details of dataset construction for each task. To generate a context of a specified length like $128k$, we randomly select multiple articles from the Noisy Context Source datasets as distractor articles. A single article is randomly chosen from Real Context Source datasets as the ground truth article. Distractor articles and the ground truth article are combined to form the whole context, ensuring that the whole context length is less than 128k and the order of all articles is shuffled. The bottom of the table contains different datasets from other benchmarks. N/A indicates that the task does not require Context Sources because the questions are synthetic rather than derived from a dataset. More details about how to construct each task are in Appendix A.2.

Task Name	Real Context Sources	Noisy Context Sources	Evaluation Metric
KV Retrieval	N/A	① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨	Accuracy
Counting Stars	N/A	① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨	Accuracy
Passage Retrieval	⑨ ⑩ ⑪ ⑫ ⑬ ⑭ ⑮	⑨ ⑩ ⑪ ⑫ ⑬ ⑭ ⑮	Accuracy
Passage Count	① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨	N/A	Accuracy
Single-doc QA	① ② ③ ④ ⑤ ⑥ ⑦ ⑧	① ② ③ ④ ⑤ ⑥ ⑦ ⑧	LLM-based Metric
Multi-doc QA	⑯ ⑰ ⑱ ⑲	① ② ③ ④ ⑤ ⑥ ⑦ ⑧	LLM-based Metric
Single-doc Sum	① ⑪ ⑫ ⑬ ⑭ ⑮	① ⑪ ⑫ ⑬ ⑭ ⑮	LLM-based Metric
Multi-doc Sum	⑳	① ⑪ ⑫ ⑬ ⑭ ⑮	LLM-based Metric

① qasper ② multifieldqa_en ③ narrativeqa ④ multidoc_qa ⑤ legal_contract_qa
⑥ financial_qa ⑦ natural_question ⑧ scientific_qa ⑨ cnn_dailymail ⑩ gov_report
⑪ qmsum ⑫ patent_summ ⑬ tv_show_summ ⑭ review_summ ⑮ meeting_summ
⑯hotpotqa ⑰2wikimqa ⑱musique ⑲rag-mini-bioasq ⑳ multi_news_e

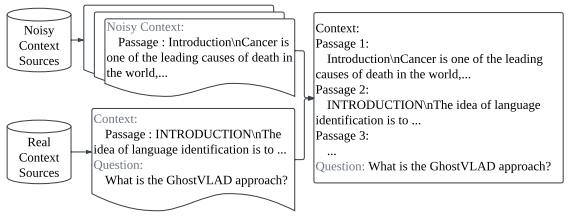

Figure 3: Illustration of the Data Generation Process for the Single-Doc QA Task

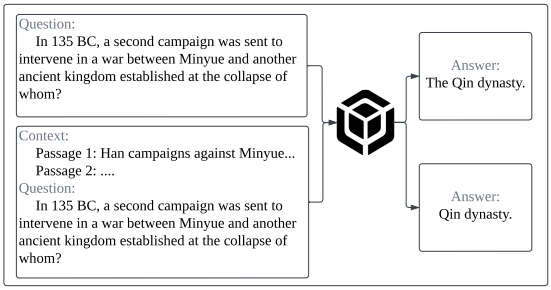

Figure 4: One sample in Question Answering where models provide accurate answers regardless of context

knowledge is often overlooked. For instance, in question-answering (QA) tasks, the model might memorize the answers to certain questions during pretraining. As shown in Figure 4, the model accurately answer questions based on its prior knowledge even without any contexts. In such cases, the model's response is not derived from the provided context but from its memorized knowledge. This oversight can lead to inflated performance metrics, misrepresenting the model's actual ability to process and comprehend long contexts. To filter out the model's prior knowledge, we introduce a QA filtering mechanism. In a no-context scenario, if the model's response score exceeds a certain threshold, it indicates that the model is relying on prior knowledge, showing the data should be excluded.

Although our length-controlled datasets are synthetically constructed, they are carefully designed to better reflect real-world usage scenarios, which we called as real-life reflective. Specifically, each instance is composed by selecting a task-relevant example as the *source* (e.g., a summarization prompt and document), and padding it with additional samples that belong to the same domain or task type (e.g., other documents suitable for summarization). This construction ensures that all components of the input are contextually aligned and task-compatible, mimicking common usage patterns in long-context settings, such as concatenated inputs in retrieval-augmented generation pipelines.

3.2 LongScore: a new long-context metric

As illustrated in Figure 2, directly using a model's scores across various text lengths to assess its long-context capability introduces inherent biases. To address this limitation, we propose a new metric that disentangles the model's base ability from its long-context capability, allowing for a more accurate and comprehensive evaluation.

Base Ability. It refers to the model's score when

Table 3: Comparison of long-context benchmarks: Longbench (Bai et al., 2023), L-Eval (An et al., 2023), ∞-Bench (Zhang et al., 2024), NIAH (Needle In A Haystack), RULER (Hsieh et al., 2024), Helmet (Yen et al., 2024), and our 💯-LongBench. L: input tokens. Controllable: The benchmark can generate contexts of specified lengths. Diverse Tasks: The tasks are varied and not limited to a single type. LLM-based Metric: Metrics in some tasks are designed based on large language models for more accurate evaluation. LC Distinction: Effectively separates the model's base ability from its long-text capability. QA Filter: Implements measures to remove the influence of the model's prior knowledge in QA tasks. The tasks in NIAH and RULER are synthetic, so they do not require LLM-based metrics or QA filtering.

Dataset	$L > 128k$	Controllable	Diverse Tasks	LLM-based Metric	LC distinction	QA Filter
Longbench	✗	✗	✓	✗	✗	✗
L-EVal	✗	✗	✓	✓	✗	✗
∞-Bench	✓	✗	✓	✗	✗	✗
NIAH	✓	✓	✗		✗	
RULER	✓	✓	✓		✗	
Helmet	✓	✓	✓	✓	✗	✗
💯-LongBench	✓	✓	✓	✓	✓	✓

conducting short-context tasks. To estimate Base Ability, we sample N instances from short text lengths (like $2k$, $4k$, $6k$). For each length, $N/3$ samples are selected, and the model's average score across these lengths is computed:

$$\text{Base Ability} = \frac{S_{2k} + S_{4k} + S_{6k}}{3} \quad (1)$$

where S_{*k} represents the performance of model with the $* - k$ length.

LongScore (LC_l) is our proposed metric. For longer lengths (e.g., 8k, 16k, 32k), we calculate the score on N instances for each length. LC_l at a given length l is then defined as:

$$\text{LC}_l = \frac{S_l - \text{Base Ability}}{\text{Base Ability}} \quad (2)$$

LongScore separates the model's Base Ability from Long-context Capability. Our metric focuses on the relative improvement or decline at longer lengths and provides a more precise assessment of long-context capabilities without being influenced by the model's Base Ability. It enables consistent and unbiased comparisons of long-context capabilities across different models and datasets.

3.3 Compare to other benchmarks

This section compares other long-context benchmarks with 💯-LongBench, highlighting their similarities and differences. The overall distinctions between benchmarks are presented in Table 3.

- LongBench (Bai et al., 2023) is an early benchmark used to evaluate the long-context capabilities. It was the first to use a variety of tasks for evaluation, but the context length is generally limited to around $8k$, and the length distribution is uneven. As many current LLMs support context lengths of 128k and beyond, these benchmarks are no longer suitable.
- ∞-Bench (Zhang et al., 2024) and L-Eval (An et al., 2023) are an improvement over benchmarks like LongBench, increasing the data length to over $128k$. However, the context length is not controllable, which limits its ability to comprehensively evaluate LLMs.
- NIAH and RULER (Hsieh et al., 2024) are designed with controllable context lengths and can control the position of the answer, specifically for evaluating long-context capabilities. These benchmarks are currently the leading tools to assess the long-context capabilities of LLMs.
- Helmet (Yen et al., 2024) is a newly proposed benchmark that not only allows for controllable context lengths but also designs a wide variety of tasks. It introduces the use of LLM-based metrics, providing a more refined way to evaluate long-context capabilities.
- 💯-LongBench generates controllable context-length tasks. Additionally, it introduces a new metric to distinguish between a model's base ability and long-context capability, offering a more comprehensive and novel approach to evaluating long-context capabilities.

4 Experimental Analysis

In this section, we conduct comprehensive experiments to first validate the reliability of 💯-LongBench and the effectiveness of the proposed

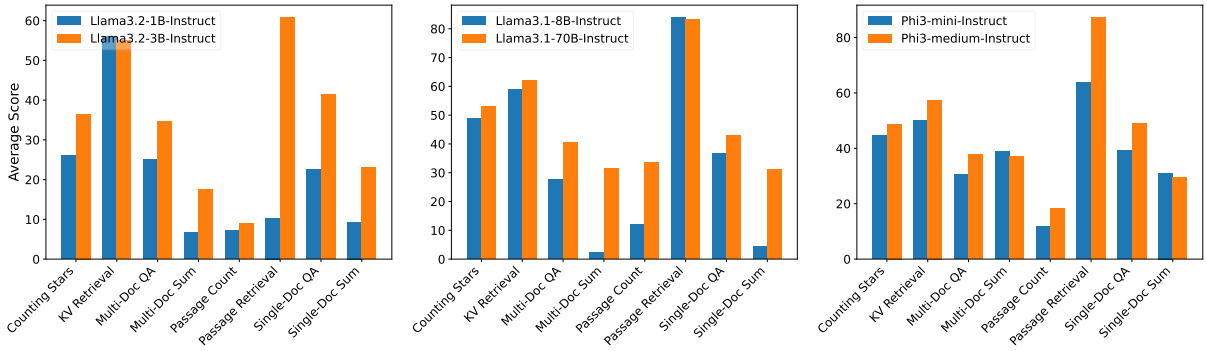

Figure 5: Verification of the reliability of 💯-LongBench: results of two models of different sizes from the same LM family tree, showcasing their average scores in different tasks. These findings confirm a well-established trend: within the same series, larger models generally outperform smaller ones, reinforcing the reliability of 💯-LongBench.

Table 4: Results of the average performance of five models across all tasks on 💯-LongBench. Base Ability represents the model's score within lengths of $2k$, $4k$ and $6k$ Avg **score** represents the average of score across lengths including $8k$, $16k$, $32k$, $64k$ and $128k$. Avg **LC** represents the average of score by using our proposed metric, LongScore. $59.1_{(1)}$ indicates that the current model has a score of 59.1 at the given context length, with a ranking of 1. Claimed Length refers to the maximum context length that the model claims to support. Qwen 2.5-14B and Qwen 2.5-7B use YaRN to extend their context length to 128k. The original context length is specified in Claimed Length.

Model	Claimed Length	Base Ability	Avg socre	Avg LC
Qwen2.5-14B-Instruct	32K	$59.1_{(1)}$	$40.7_{(1)}$	$-31.1_{(4)}$
Qwen2.5-7B-Instruct	32K	$57.4_{(2)}$	$39.8_{(2)}$	$-30.6_{(3)}$
Llama3.1-8B-Instruct	128K	$44.0_{(3)}$	$36.3_{(3)}$	$-17.4_{(1)}$
Llama3.2-1B-Instruct	128K	$28.7_{(4)}$	$20.4_{(4)}$	$-28.8_{(2)}$

metric. They are then used to evaluate the long-context capabilities of several open-source models.

4.1 Verification of the reliability of the proposed benchmark

To verify the reliability of 💯-LongBench, we evaluate three model families (Llama 3.2, Llama 3.1, and Phi 3), selecting two different model sizes from each family. Since these are models of different sizes within the same series, the expected trend in the dataset would be: for the same series, larger models generally perform better in all tasks across different context lengths. As shown in Figure 5, this overall trend is observed, indicating that the dataset generation itself is reliable and can be used for evaluating long-context capabilities. For instance, compare to Llama 3.2-1B-Instruct, Llama 3.2-3B-Instruct gets higher average scores in each task. For more detailed results of models across various context lengths, refer to Appendix A.4.

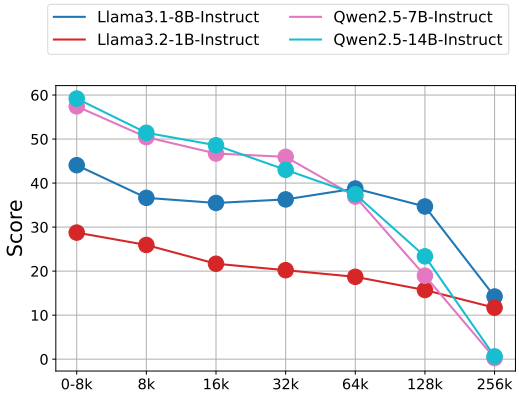

Figure 6: Results of four open-source models on all tasks in 💯-LongBench, showing their average scores of all eight tasks at different context lengths.

4.2 Verification of the effectiveness of the proposed metric

Following the setting of Lu et al. (2024), we compare two long-context enhancement methods, NTK and PI, using LongBench and 💯-LongBench. On 💯-LongBench, we evaluate performances with two metrics: score and LongScore (LC). We include three evaluations to further validate the discriminative power and practical value of our proposed LongScore metric. These comparisons were chosen to reflect real-world modeling choices and align with community intuition: (1) NTK vs. PI on long-context tasks, (2) performance of LLaMA3-8B-Instruct under different RoPE theta ratios, and (3) Gemini-1.5 model variants like Gemini-1.5-Flash and Gemini-1.5-Pro from HEMLET benchmark.

There are some reasons why we choose these three comparisons: (1) NTK and PI are chosen for comparison because it is well-established that NTK provides a more fine-grained extension of PI. (2) We choose LLaMA 3-8B-Instruct (8k claimed context length) with different RoPE theta ratios. Generally speaking, appropriately increasing the RoPE

Table 5: Results of 4 models' ranking in Ruler(Hsieh et al., 2024) on different metrics. **Base Ability** represents the model's score with a 4k-length context. Avg represents the average of scores excluding the base score. $95.8_{(1)}$ indicates that the current model has a score of 95.8 at the given context length, with a ranking of 1. LC represents the score by our proposed metric, LongScore.

Models	Claimed Length	Base Ability	$8k$		$16k$		$32k$		$64k$		$128k$		Avg	
			score	LC	score	LC	score	LC	score	LC	score	LC	score	LC
Llama3.1 (70B)	128K	$96.5_{(1)}$	$95.8_{(1)}$	$-0.7_{(2)}$	$95.4_{(1)}$	$-1.1_{(1)}$	$94.8_{(1)}$	$-1.7_{(1)}$	$88.4_{(1)}$	$-8.3_{(1)}$	$66.6_{(2)}$	$-30.9_{(3)}$	$88.2_{(1)}$	$-8.6_{(2)}$
Yi (34B (Young et al., 2024))	200K	$93.3_{(2)}$	$92.2_{(3)}$	$-1.1_{(3)}$	$91.3_{(2)}$	$-2.1_{(2)}$	$87.5_{(2)}$	$-6.2_{(2)}$	$83.2_{(2)}$	$-10.8_{(2)}$	$77.3_{(1)}$	$-17.1_{(1)}$	$86.3_{(2)}$	$-7.5_{(1)}$
Phi3-medium (14B)	128K	$93.3_{(3)}$	$93.2_{(2)}$	$-0.1_{(1)}$	$91.1_{(2)}$	$-2.3_{(3)}$	$86.8_{(3)}$	$-6.9_{(3)}$	$78.6_{(3)}$	$-15.7_{(3)}$	$46.1_{(4)}$	$-50.5_{(4)}$	$79.1_{(3)}$	$-15.1_{(4)}$
LWM (7B) (Liu et al., 2024a)	1M	$82.3_{(4)}$	$78.4_{(4)}$	$-4.70_{(4)}$	$73.7_{(4)}$	$-10.4_{(4)}$	$69.1_{(4)}$	$-16.0_{(4)}$	$68.1_{(4)}$	$-17.2_{(4)}$	$65.0_{(3)}$	$-21.0_{(2)}$	$70.8_{(4)}$	$-13.9_{(3)}$

Table 6: **Comparison of models and methods under our proposed LongScore metric.** We present three evaluations to validate the discriminative power of LongScore: (1) NTK vs. PI on 100-LongBench; (2) LLaMA3-8B with different RoPE theta ratios; (3) Gemini-1.5 variants from the HEMLET benchmark. In all cases, LongScore reflects performance differences that align with common understanding (e.g., NTK > PI, Gemini-Pro > Gemini-Flash), while amplifying meaningful gaps that are not visible with raw accuracy. The results highlight the discriminative ability and effectiveness of our proposed benchmark and metric.

Benchmark	Model / Method	base	8k	16k	24k / 32k	48k / 64k	128k / 256k	avg(score)	avg(LongScore)
100-LongBench	PI	19.18	16.47	17.67	17.10	17.67	0.44	13.87	-27.68
	NTK	19.39	15.72	16.53	16.70	17.17	12.88	15.83	-18.40
100-LongBench	LLaMA3-8B (ratio=1)	35.37	37.08	1.45	1.87	0.52	0.99	7.13	-79.84
	LLaMA3-8B (ratio=64)	32.52	31.94	25.34	26.08	26.94	1.63	18.83	-42.12
HEMLET	Gemini-1.5-Flash	59.6	–	60.2	58.1	55.0	50.7	56.00	-6.04
	Gemini-1.5-Pro	59.5	–	60.1	59.9	57.0	54.1	57.77	-2.90

theta improves the model's long context capability (within a reasonable extent). (3) we choose Gemini-1.5-Flash and Gemini-1.5-Pro because they have an obvious difference in long-context ability.

On the LongBench tasks, both NTK and PI methods perform similarly, failing to provide a clear distinction. However, as shown in Table 6, on 💯-LongBench, the differences between NTK and PI became much more apparent across the selected tasks, effectively differentiating the two methods. Moreover, it is obvious that the differences of NTK and PI measured by LongScore are greater than those measured by the traditional metric, showing that LongScore demonstrates a greater ability to highlight these differences compared to the traditional metric and a more effective tool for distinguishing long-context capabilities.

In other pairwise comparison, LongScore readings show a much more pronounced difference compared to the original scoring metrics of the datasets, while the win-loss order remains consistent with our general understanding of a model or method's long context capability (NTK > PI, ratio=64 > ratio=1, Gemini-1.5-Pro > Gemini-1.5-Flash). These results highlight the discriminative power and effectiveness of our LongScore.

4.3 Experiments on frontier open-source LLMs

This section introduces the experiments conducted using 💯-LongBench and the proposed metric, aimed at evaluating the long-context capabilities of various popular open-source large models.

We select four models, due to GPU resource limitations, as they can be used to generate outputs with a $256k$ context length. For each of the eight tasks, we generated 100 samples at each context length ($8k$, $16k$, $32k$, $64k$, $128k$) to obtain the scores, using the performance at $2k$, $4k$, and $6k$ as Base Ability. Finally, the average scores across all tasks are computed. Table 4 presents average results and the corresponding rankings. Figure 6 displays average scores at each context length.

Here we explain why we choose the appropriate context lengths (e.g. 2k, 4k, 6k) for measuring Base Ability. We evaluate 8 models spanning the `LLaMA 3.1`, `Phi-3`, and `Qwen 2.5` families. These models typically undergo pretraining with context lengths of 4K or 8K tokens before undergoing further continuous pretraining for long-context extension. Given this, we generalize that most models in our study have a pre-extension context window of either 4K or 8K. To probe their base reasoning ability, we evaluate performance under 2K, 4K, and 6K context lengths. These values are chosen to provide representative coverage of the model's original pretraining range without exceeding it, thereby offering a stable measure of Base Ability.

Interestingly, as shown in Table 4, the rankings obtained by the traditional metric are almost identical to the rankings based on Base Ability.

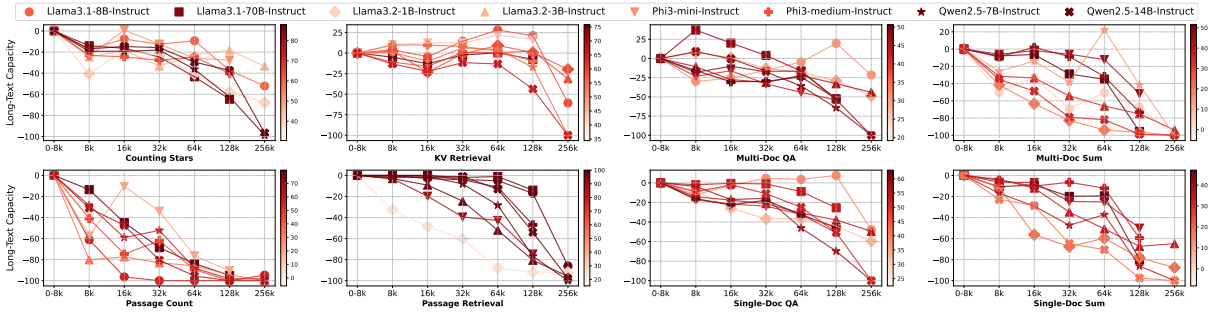

Figure 7: Results of eight open-source models on eight tasks are presented, with their scores calculated using LongScore metric. Each markrer represents a single model. The darker the color of the line, the stronger the base ability of the model.

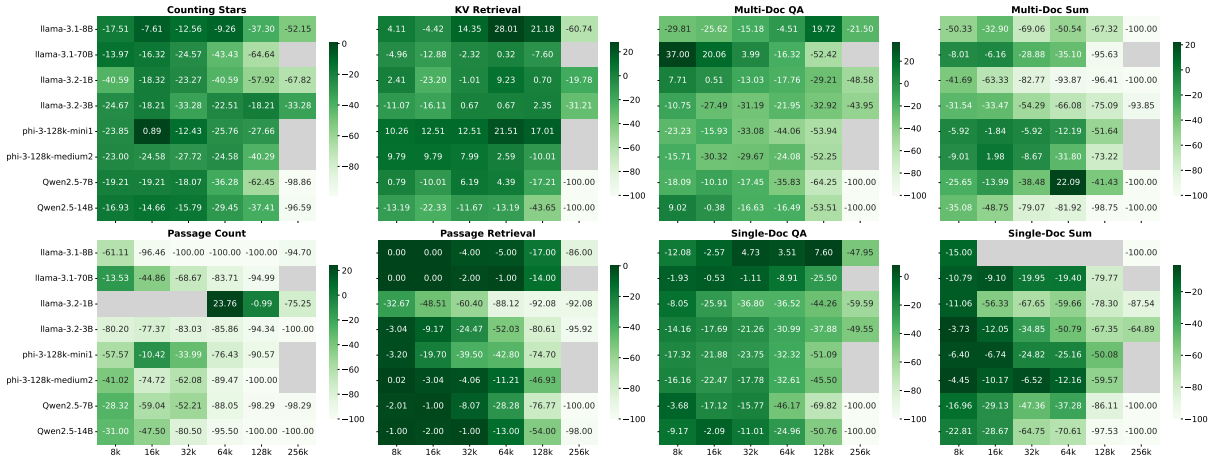

Figure 8: Results of eight models on 💯-LongBench by using LongScore metric. The gray shading indicates either anomalous models' scores or cases where the model is unable to generate outputs for $256k$-long contexts.

However, rankings using LongScore metric show a significant difference from Base Ability rankings, as demonstrated by models like Qwen 2.5-14B-Instruct and Qwen 2.5-7B-Instruct. From Figure 6, it can be observed that while these two models have higher scores at shorter context lengths (e.g. 8k, 16k), their scores drop significantly at longer context lengths (128k, 256k). This indicates that current long-text evaluation metrics are heavily influenced by Base Ability, while LongScore (the metric proposed in this paper) separates base ability from long-context capability, providing a more accurate reflection of the model's long-context performance. For comparisons of more open-source models on 💯-LongBench and their long-context capability evaluation, please refer to Appendix A.5.

We also present the results of eight models from four LLM family trees (Llama 3.1, Llama 3.2, Qwen 2.5 and Phi 3) on 💯-LongBench. The evaluation uses LongScore metric and the detailed results about each task are shown in Figure 7 and Figure 8.

Long-context ability is important in certain specialized domains such as healthcare and law. To this end, we additionally include several domain-specific long-context tasks, including Medical-Summary, MedOdyssey (Fan et al., 2024), and CaseSumm (Heddaya et al., 2024). We re-evaluate the performance of the `LLaMA 3.2-1B-Instruct` model with and without these datasets. The detailed results are shown in Appendix A.6.

4.4 Experiments on Ruler with different metrics

We utilize data from Ruler (Hsieh et al., 2024), using a $4k$-length context to represent the model's base ability. The results are shown in Table 5, where we evaluate four models' performance at different context lengths using both LongScore and the traditional metric. Compared to LLaMA 3.1 (70B), Yi (34B) (Young et al., 2024) has a slightly lower overall score before reaching 128k context length, but at 128k, Yi (34B) performs significantly better. Similarly, compared to Phi3-medium (14B), LWM (7B) shows lower base ability and shorter text handling but clearly outperforms Phi3-medium at 128k. If ranking is based solely on scores,

LLaMA 3.1 (70B) and Phi3-medium (14B) would be ranked higher than their counterparts, but this does not show their true long-context capabilities. By using LongScore, we correct this discrepancy.

5 Related Works

In this section, we review relevant prior research connected to our study. We summarize cutting-edge models known for their strong long-text processing capabilities, explore methods designed to enhance these abilities, and examine the benchmarks commonly used to assess long-text proficiency. Additionally, wwe discuss the limitations of existing benchmarks, not disentangling Base Ability from true long-context capabilities.

Long-context language models. Both open-source and closed-source state-of-the-art models now support extended context lengths of up to 128K tokens or more, including GPT-4 (Achiam et al., 2023), Gemini (Team et al., 2024), Claude (Caruccio et al., 2024), LLaMA-3 (Dubey et al., 2024), and Phi-3 (Abdin et al., 2024). These models typically achieve long-context capabilities through a combination of improved pretraining and post-training techniques. For instance, many models adopt two-stage or continued pretraining pipelines, where an initial short context window (e.g., 4K or 8K) is later extended to longer lengths (e.g., 128K) using scalable attention mechanisms such as FlashAttention (Dao et al., 2022) and optimized positional encoding schemes (Li et al., 2021; Xiong et al., 2023; Hsu et al., 2024). This trend is well-documented in recent technical reports (Yang et al., 2024; Abdin et al., 2024; Dubey et al., 2024), which highlight how careful adjustments to training schedules, data distribution, and architecture design contribute to stable performance in extreme long-context settings. Nonetheless, despite these advancements, effectively evaluating and comparing the true reasoning ability of such models in long-context scenarios remains a significant challenge in the real situations and scenarios.

Long context methods. Many studies have explored methods to extend the context window length of models during fine-tuning, with some approaches even achieving this without fine-tuning. Techniques such as Position interpolation (PI) (Chen et al., 2023a), NTK (Peng and Quesnelle, 2023), YaRN (Peng et al., 2023) and SelfExtend (Jin et al., 2024) manipulate RoPE (Rotary Position Embedding) (Su et al., 2024) to do length extension. Other methods, including Retrievers (Xu et al., 2023), StreamingLLM (Xiao et al., 2023b), LM-Infinite (Han et al., 2024), Longlora (Chen et al., 2023b), Inf-LLM (Xiao et al., 2024) and Landmark (Mohtashami and Jaggi, 2023), focus on designing new attention architectures or exploiting specific phenomena in attention mechanisms (Sun et al., 2024) to achieve length extension. Additionally, some works (Jiang et al., 2023; Li et al., 2023b) focus on reducing length extension to length compression via a summarization step, where long contexts are compressed or summarized before being processed by the model.

Long-context benchmarks. LongBench (Bai et al., 2023) and L-Eval (An et al., 2023) are early benchmarks for evaluating long-context capabilities. Later benchmarks, such as ∞-Bench (Zhang et al., 2024), extended the context length of datasets further. Subsequently, synthetic task-related benchmarks like NIAH(Needle In A Haystack), and Ruler (Hsieh et al., 2024) emerged, focusing not only on evaluating contextual capabilities but also on examining models' sensitivity to the positional appearance of text. More recently, benchmarks such as HELMET (Yen et al., 2024) and LV-Eval (Yuan et al., 2024) introduced controllable context lengths and LLM-based metrics. Building on them, this work further considers prior model knowledge, and introduces a novel metric.

6 Conclusion

Our benchmark and metric address key shortcomings in current evaluation methodologies, such as the inability to isolate long-context reasoning from baseline performance and reliance on insufficiently representative tasks. By incorporating real-world data, diverse task types and difficulties, and a novel metric (LongScore), 💯-LongBench provides a robust platform to evaluate and compare LLMs across varying context lengths. This allows for a deeper understanding of how models handle extended contexts while minimizing the influence of prior knowledge or base abilities. As LLMs continue to evolve, the ability to rigorously assess their long-context reasoning will play a critical role in identifying bottlenecks and guiding the design of next-generation models. Our approach sets a new standard for assessing LLMs, paving the way for more robust innovations in long-context evaluation. Furthermore, it will provide an actionable insight for optimizing model architectures and training strategies to enhance long-context capabilities.

Limitations

The proposed metric requires models to demonstrate relatively strong base ability on the task. If a model's base ability is insufficient, subsequent evaluations of long-context capabilities may exhibit significant fluctuations, making it less effective for comparing models' long-context performance. Besides, when constructing the benchmark, it is necessary to select articles of varying lengths to assemble into noisy contexts. For shorter target lengths, such as 2k tokens, the selected articles should also have shorter lengths — preferably less than 1k tokens — to ensure the context can be formed with two or more documents. Therefore, it is essential to collect texts of diverse lengths, particularly shorter ones, to enable effective assembly of the desired contexts.

Acknowledgements

This research was partially supported by NSF Awards OAC-2117439. Further, this work made use of the High Performance Computing Resource in the Core Facility for Advanced Research Computing at Case Western Reserve University (CWRU). We give special thanks to the CWRU HPC team for their prompt and professional help and maintenance. The views and conclusions in this paper are those of the authors and do not represent the views of any funding or supporting agencies.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Rishabh Agarwal, Avi Singh, Lei M Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, et al. 2024. Many-shot in-context learning. *arXiv preprint arXiv:2404.11018*.

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

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

Loredana Caruccio, Stefano Cirillo, Giuseppe Polese, Giandomenico Solimando, Shanmugam Sundaramurthy, and Genoveffa Tortora. 2024. Claude 2.0 large language model: Tackling a real-world classification problem with a new iterative prompt engineering approach. *Intelligent Systems with Applications*, 21:200336.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023a. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023b. Longlora: Efficient fine-tuning of long-context large language models. *arXiv preprint arXiv:2309.12307*.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yongqi Fan, Hongli Sun, Kui Xue, Xiaofan Zhang, Shaoting Zhang, and Tong Ruan. 2024. Medodyssey: A medical domain benchmark for long context evaluation up to 200k tokens. *Preprint*, arXiv:2406.15019.

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. 2024. How to train long-context language models (effectively). *arXiv preprint arXiv:2410.02660*.

Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. 2024. Lm-infinite: Zero-shot extreme length generalization for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3991–4008.

Mourad Heddaya, Kyle MacMillan, Anup Malani, Hongyuan Mei, and Chenhao Tan. 2024. Casesumm: A large-scale dataset for long-context summarization from u.s. supreme court opinions. *Preprint*, arXiv:2501.00097.

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

Pin-Lun Hsu, Yun Dai, Vignesh Kothapalli, Qingquan Song, Shao Tang, Siyu Zhu, Steven Shimizu, Shivam Sahni, Haowen Ning, and Yanning Chen. 2024. Liger kernel: Efficient triton kernels for llm training. *arXiv preprint arXiv:2410.10989*.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. *arXiv preprint arXiv:2310.06839*.

Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. 2024. Llm maybe longlm: Self-extend llm context window without tuning. *Preprint*, arXiv:2401.01325.

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

Shenggui Li, Fuzhao Xue, Chaitanya Baranwal, Yongbin Li, and Yang You. 2021. Sequence parallelism: Long sequence training from system perspective. *arXiv preprint arXiv:2105.13120*.

Yucheng Li, Bo Dong, Chenghua Lin, and Frank Guerin. 2023b. Compressing context to enhance inference efficiency of large language models. *arXiv preprint arXiv:2310.06201*.

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. 2024. Chain of thought empowers transformers to solve inherently serial problems. *arXiv preprint arXiv:2402.12875*.

Hao Liu, Wilson Yan, Matei Zaharia, and Pieter Abbeel. 2024a. World model on million-length video and language with blockwise ringattention. *CoRR*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024b. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Yi Lu, Jing Nathan Yan, Songlin Yang, Justin T Chiu, Siyu Ren, Fei Yuan, Wenting Zhao, Zhiyong Wu, and Alexander M Rush. 2024. A controlled study on long context extension and generalization in llms. *arXiv preprint arXiv:2409.12181*.

Amirkeivan Mohtashami and Martin Jaggi. 2023. Landmark attention: Random-access infinite context length for transformers. *arXiv preprint arXiv:2305.16300*.

Bowen Peng and Jeffrey Quesnelle. 2023. Ntk-aware scaled rope allows llama models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation.

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

Mingyang Song, Mao Zheng, and Xuan Luo. 2024. Counting-stars: A multi-evidence, position-aware, and scalable benchmark for evaluating long-context large language models. *Preprint*, arXiv:2403.11802.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. 2024. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Chonghua Wang, Haodong Duan, Songyang Zhang, Dahua Lin, and Kai Chen. 2024. Ada-leval: Evaluating long-context llms with length-adaptable benchmarks. *Preprint*, arXiv:2404.06480.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. 2024. Infllm: Training-free long-context extrapolation for llms with an efficient context memory. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023a. Efficient streaming language models with attention sinks. *arXiv*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023b. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, et al. 2023. Effective long-context scaling of foundation models. *arXiv preprint arXiv:2309.16039*.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2023. Retrieval meets long context large language models. *arXiv preprint arXiv:2310.03025*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

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

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. 2024. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*.

Tao Yuan, Xuefei Ning, Dong Zhou, Zhijie Yang, Shiyao Li, Minghui Zhuang, Zheyue Tan, Zhuyu Yao, Dahua Lin, Boxun Li, et al. 2024. Lv-eval: A balanced long-context benchmark with 5 length levels up to 256k. *arXiv preprint arXiv:2402.05136*.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, et al. 2024. Bench: Extending long context evaluation beyond 100k tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15262–15277.

A Appendix

A.1 Results of models' long-text enhancement methods on Longbench

These section introduces four long-context enhancement method's performances on three LongBench tasks. The colored dashed lines represent the average score of each model on the corresponding task. The size of the markers corresponds to the proportion of each text length within the entire dataset. The larger the marker, the higher the proportion. The results exhibit significant variation across tasks of different lengths within the same dataset. All results are in Appendix A.1.

A.2 Details about how to construct each task

KV Retrieval. This task primarily evaluates the model's ability to extract critical information while ignoring irrelevant content and noisy information. (1) Context Construction: Three pairs of key-value (k_1, v_1; k_2, v_2; k_3, v_3) are generated using UUIDs. The value of the previous pair serves as the key for the subsequent pair ($v_1 = k_2$; $v_2 = k_3$). These key-value pairs are randomly inserted into different noisy contexts. The noise introduces irrelevant or distracting information, simulating real-world challenges. (2) Question Setup: The question asks the model to identify the value corresponding to a specific key. (3) Evaluation Metric: The task is evaluated using accuracy (Acc). If the model correctly identifies the value associated with the queried key, its accuracy score is incremented by one.

Counting Stars. Following (Song et al., 2024) , this task assesses the model's ability to extract critical information across multiple documents, maintain the correct sequence when aggregating information and resist distractions from misleading or altered options. (1) Context Construction: Four noisy context passages are selected from all noisy context passages and each passage is appended with a sentence in the format: *The little penguin counted* N ★, where N represents a specific number of stars counted in that passage. (2) Question Setup: The model is tasked with identifying the sequence of star counts in the order of sentence appearance, like [38, 10, 90, 42]. The task provides multiple-choice options, including the correct sequence and several distractors. Distractors are generated by swapping numbers, modifying values, or changing the order to increase difficulty. (3) Evaluation Metric: The task is evaluated using accuracy (Acc). If the model selects the correct sequence, its accuracy score is incremented by one.

Passage Retrieval. By focusing on comprehension and recognition, this task challenges the model's ability to extract and correlate key information in a multi-document setting. (1) Context Construction: A single data sample comprises multiple articles, each sourced from a distinct domain. These articles are concatenated to form the context. (2) Question Setup: The model is provided with the summary of one specific article from the context. The task is to identify which article in the context corresponds to the given summary. (3) Evaluation Metric: The task is evaluated using accuracy (Acc). If the model correctly identifies the article corresponding to the summary, its accuracy score is incremented by one.

Passage Count. The task assesses a model's ability to understand and integrate global key information by determining the number of unique articles within a multi-article context. (1) Context Construction: Each data sample comprises multiple articles sourced from different domains. Some articles are repeated multiple times within the context to add redundancy and complexity. (2) Question Setup: The model is tasked with identifying the total number of unique (non-repeated) articles in the context. (3) Evaluation Metric: The task is evaluated using accuracy (Acc). If the model correctly identifies the count of unique articles, its accuracy score is incremented by one.

Single-Doc QA. The task evaluates a model's ability to answer questions specific to a single article within a multi-article context. (1) Context Construction: Each data sample consists of multiple articles from different domains. A specific question is posed about one particular article within the context. (2) Evaluation Metric: The model's answers are assessed using another large language model (like GPT-4o-mini). Evaluation is based on two dimensions: Fluency is scored on a 3-point scale (0, 1, 2), evaluating the coherence and readability of the answer. Correctness is scored on a 4-point scale (0, 1, 2, 3), assessing the factual accuracy of the response in relation to the context. The final score is calculated as the product of the Fluency and Correctness scores: Final Score $=$ Fluency $\times$ Correctness (3) Prior Knowledge Filtering: To filter out the model's prior knowledge, we introduce a filtering process. In a no-context scenario, if the model's response score exceeds a certain threshold, it indicates that the

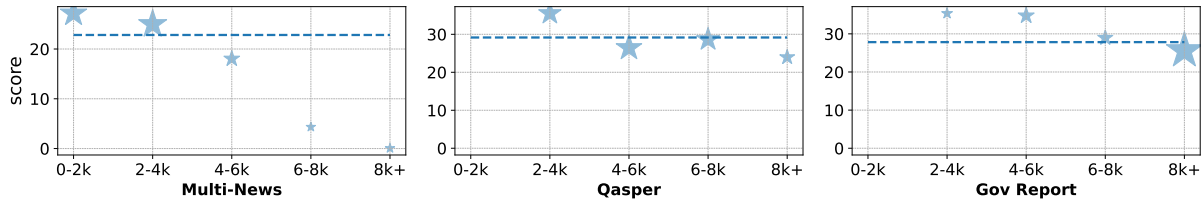

Figure 9: Illustration of NTK's performances on three LongBench tasks.

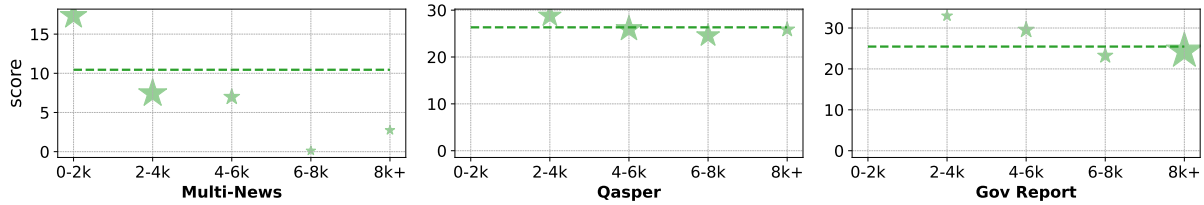

Figure 10: Illustration of PI's performances on three LongBench tasks.

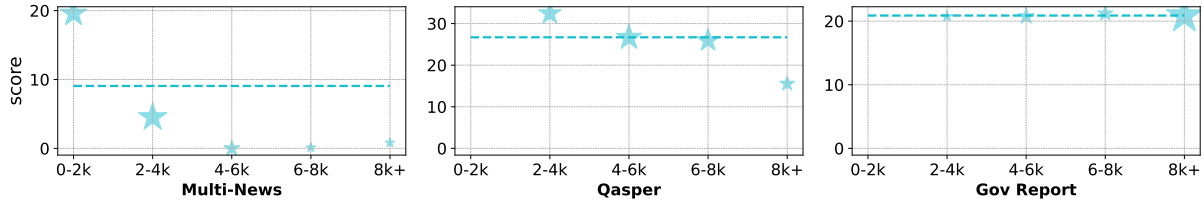

Figure 11: Illustration of YaRN's performances on three LongBench tasks.

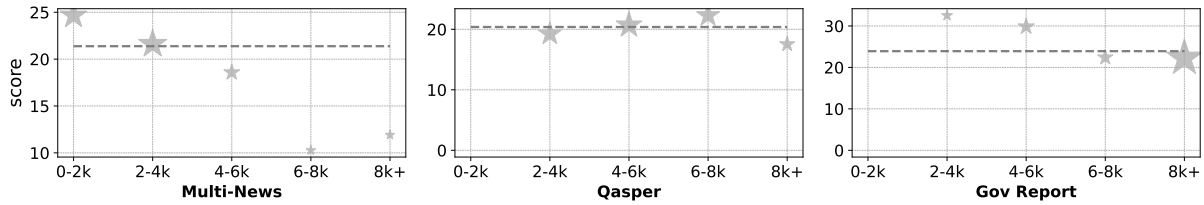

Figure 12: Illustration of Longlora's performances on three LongBench tasks.

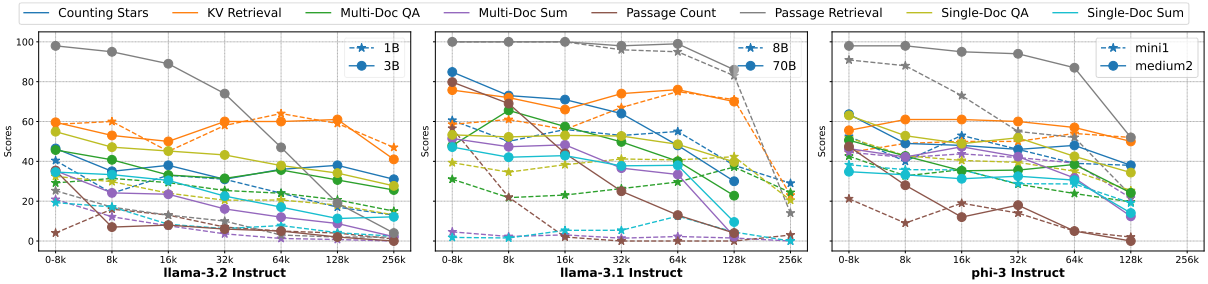

Figure 13: Verification the reliability of 💯-LongBench: results of two models of different sizes from the same LM family tree, showcasing their scores in different tasks across various context lengths. One color represents a specific task, with solid lines indicating larger models and dashed lines representing smaller models. The results of different LMs from the same LM family tree basically validate the general trend: the larger model tend to get a higher score while the score decreases as the context length increases.

model is relying on prior knowledge. In such cases, the data is excluded from the statistical analysis.

Multi-Doc QA. The task evaluates a model's ability to integrate information from multiple articles and provide coherent, accurate answers to questions that require a global understanding of the context. (1) Context Construction: Each data sample contains multiple articles from different domains. The question posed requires the model to synthesize information across multiple articles to generate the correct answer. (2) Evaluation Metric: Similar to the Single-Doc QA task, the model's answers are evaluated using another large language model and evaluated by the same dimensions. (3) Prior Knowledge Filtering is similar to the Single-Doc QA task.

Single-Doc Sum. The task evaluates a model's ability to generate concise and accurate summaries for a specific article within a multi-article context. (1) Context Construction: Each data sample consists of multiple articles from different domains. (2) Question Setup: The model is tasked with summarizing the content of one specific article from the context. (3) Evaluation Metric: The generated summary is assessed by another large language model. Two scoring dimensions are considered: Fluency evaluates the coherence and readability of the summary and is scored on a 2-point scale: 0 (poor fluency), 1 (good fluency). Precision measures the relevance of the summary by comparing each sentence in the model's output to the reference summary. and is calculated as $\text{Precision} = \frac{\text{Number of relevant sentences}}{\text{Total number of sentences in the summary}}$. The final score is the product of these two dimensions: $\text{Final Score} = \text{Fluency} \times \text{Precision}$. By requiring accurate and readable summaries, this task emphasizes the model's capacity for effective information synthesis and integration.

Multi-Doc Sum. The task evaluates a model's ability to integrate information from multiple articles and produce a coherent and accurate summary of their shared content. (1) Context Construction: Each data sample consists of multiple articles from different domains. (2) Question Setup: The model is tasked with summarizing the relevant content from all provided articles. (3) Evaluation Metric: Similar to the Single-Doc Sum task, the model's answers are evaluated using another large language model and evaluated by the same dimensions. By requiring effective summarization of multi-document content, this task highlights the model's ability to synthesize and generalize information across diverse sources.

A.3 Prompts used in each task

This section presents the prompts used in each task. Here, *{context}* represents the entire context constructed from articles in the noisy context sources and real context sources. *{input}* represents the question for the task, and *{instruction}* represents the model-specific instructions. For example, in Single-Doc QA, the instruction might be "Answer the question related to Passage 1", indicating that the question is specifically based on Passage 1.

KV Retrieval. *There are some passages below sourced from many different fields.\n\n {context} \n\n Given several key-value pairs in these passages, you need to find the value of the key. Read the question related with these key-value pairs and give the correct answer. {input}*

Counting Stars. *There are some passages below sourced from many different fields.\n \n {context} \n\n Only output the results without any explanation. Read the following question and give the correct answer: {input} \n The final answer is:*

Passage Retrieval. *Here are some passages from many different fields, along with an summarization. Please determine which passage the summarization is from.\n \n {context} \n \n The following is a summarization.\n\n {input} \n \n Please enter the number of the passage that the summarization is from. The answer format must be like "Passage 1", "Passage 2", etc. \n\n The answer is Passage*

Passage Count. *There are some paragraphs below sourced from many different fields. Some of them may be duplicates. Please carefully read these paragraphs and determine how many unique paragraphs there are after removing duplicates. In other words, how many non-repeating paragraphs are there in total? \n\n {context} \n\n Please enter the final count of unique paragraphs after removing duplicates. The output format should only contain the number, such as 1, 2, 3, and so on.\n\n The final answer is:*

Single-Doc QA. *Answer the question based on the given passages. Only give me the answer and do not output any other words.\n \n The following are given passages and these passages are from many different fields.\n \n {context} \n \n Answer the question based on the given passages following the instruction: \n {instruction} \n \n Question: {input} \n Only give me the answer and do not output any other words. Answer: \n",*

Multi-Doc QA. *Answer the question based on the given passages. Only give me the answer and do not output any other words.\n \n The following are given passages and these passages are from many different fields.\n \n {context} \n \n Answer the question based on the given passages following the instruction: \n {instruction} \n \n Question: {input} \n Only give me the answer and do not output any other words. Answer: \n*

Single-Doc Sum. *You are given several passages as follows, but not all of them need to be summarized. \n \n {context} \n \n Please follow these instructions: \n 1.{input} \n 2.Ignore and do not summarize any passages not listed above. \n 3.For the selected passages, the summary should include: the main arguments or conclusions of each article, the key evidence or supporting data presented and any unique or innovative points made in the passages. \n 4.The summary should be concise, focusing only on the most important information from the passages. Now, please generate the summary for the specified passage, following the instructions carefully. \n Summary: \n*

Multi-Doc Sum. *You are given several passages as follows, but not all of them need to be summarized.\n \n {context} \n \n Please follow these instructions:\n 1.{input} \n 2.Ignore and do not summarize any passages not listed above. \n 3.All the selected passages should be summarized into a few short sentences and do not summarize each selected passages separately. The summary should include: the main arguments or conclusions of each article, the key evidence or supporting data presented and any unique or innovative points made in the passages. \n 4.The summary should be concise, focusing only on the most important information from the passages. Now, please combine and summarize the main ideas from the selected relevant passages into one cohesive summary, following the instructions carefully.\n \n Summary: \n*

A.4 Further verification of the reliability of the proposed benchmark

To further verify the reliability of the generated dataset, we evaluate three model families (Llama 3.2, Llama 3.1, and Phi 3), selecting two different model sizes from each family. Given that these models are from the same series but vary in size, the expected trends on the dataset are as follows: (1) Model Size Effect: Larger models should generally achieve higher scores compared to smaller models within the same series. (2) Text Length Effect: As the text length increases, the performance scores should decrease across all models. As shown in Figure 13, the results basically follow these expected trends: larger models tend to score higher, and performance decreases as text length increases. This consistent pattern indicates that the dataset generation process is accurate and reliably.

Table 7: Results of the average performance of five models across all tasks on 💯-LongBench. **Base Ability** represents the model's score within lengths of $2k$, $4k$ and $6k$ *Avg* **score** represents the average of score across lengths including $8k$, $16k$, $32k$, $64k$ and $128k$. *Avg* **LC** represents the average of score by using our proposed metric. $57.4_{(1)}$ indicates that the current model has a score of 57.4 at the given context length, with a ranking of 1. Claimed Length refers to the maximum context length that the model claims to support. Qwen 2.5-14B and Qwen 2.5-7B use YaRN to extend their context length to 128k. so, the original context length is specified in Claimed Length.

model	Claimed Length	Base Ability	*Avg* score	*Avg* LC
llama-3.1-70B-Instruct	128K	$67.5_{(1)}$	$52.55_{(1)}$	$-22.18_{(2)}$
Qwen2.5-14B-Instruct	32K	$59.1_{(2)}$	$40.77_{(3)}$	$-31.12_{(7)}$
Phi-3-128k-medium	128K	$57.4_{(3)}$	$43.28_{(2)}$	$-24.65_{(4)}$
Qwen2.5-7B-Instruct	32K	$57.4_{(4)}$	$39.80_{(4)}$	$-30.69_{(6)}$
Llama3.2-3B-Instruct	128K	$51.2_{(8)}$	$34.81_{(7)}$	$-32.06_{(8)}$
Phi-3-128k-mini	128K	$48.2_{(6)}$	$36.78_{(5)}$	$-23.85_{(3)}$
Llama-3.1-8B-Instruct	128K	$44.0_{(7)}$	$36.37_{(6)}$	$-17.46_{(1)}$
Llama3.2-1B-Instruct	128K	$28.7_{(8)}$	$20.45_{(8)}$	$-28.88_{(5)}$

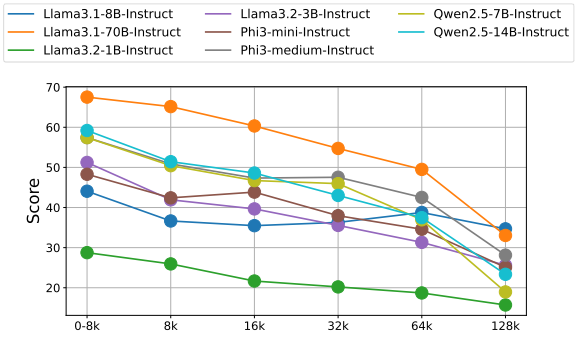

Figure 14: Results of eight open-source models on all tasks in 💯-LongBench, showing their scores at different context lengths.

A.5 Results of different Open-source models on our proposed benchmark

This section first introduces the experiments conducted using 💯-LongBench and the proposed metric, aimed at evaluating the long-context capabilities of various popular open-source large models.

We select eight open-source models. For each of the eight tasks, we generated 100 samples at each context length ($8k$, $16k$, $32k$, $64k$ and $128k$)

Table 8: Performance of `LLaMA 3.2-1B-Instruct` with and without domain-specific tasks. We report scores across different context lengths and two average metrics: overall average and average on long contexts (32k+). Adding healthcare and law tasks leads to a slight drop in average long-context performance.

Benchmark	base	8k	16k	32k	64k	128k	avg(score)	avg(LongScore)
original	24.41	22.42	20.55	18.54	17.92	15.44	18.97	-22.27
original + healthcare & law	24.58	21.97	18.49	15.77	16.64	12.83	17.14	-30.27

to obtain the scores. The model's Long-context Capability was then calculated, using the performance at $2k$, $4k$, and $6k$ as the base ability. Finally, the average scores across all tasks for the five models are computed. Table 7 presents the final average results and the corresponding rankings of the five models. Figure 14 displays the average scores for all tasks at each context length for the five models.

A.6 Results of models with and without domain-specific tasks

We have added long text datasets from the recommended domains (law and healthcare) to enhance the comprehensiveness of our benchmark. Evaluating the capability of LLMs to handle such domain-specific scenarios is indeed a crucial need.

Specifically, we mix up CaseSumm, MedOdyssey, and Medical Summary into our original dataet. We reevaluate the performance of the LLaMA 3.2 1B-Instruct model with and without such datasets.

As is shown in Table 8, incorporating healthcare and law-focused domain-specific data leads to a slight performance decline in long text scenarios, likely because the model lacks comprehensive knowledge in these specialized fields. However, the overall trend is steady. We plan to incorporate this additional evaluation to our updated manuscript and add more discussion regarding domain-specific long context evalutions.