LongLeader: A Comprehensive Leaderboard for Large Language Models in Long-context Scenarios

Pei Chen[1] **Hongye Jin**[2] **Cheng-Che Lee**[1] **Rulin Shao**[3] **Jingfeng Yang**[1]
Mingyu Zhao[1] **Zhaoyu Zhang**[1] **Qin Lu**[1] **Kaiwen Men**[3] **Ning Xie**[1]
Huasheng Li[1] **Bing Yin**[1] **Han Li**[1] **Lingyun Wang**[1]
[1]Amazon
[2]Texas A&M University
[3]University of Washington
{ppeichen, leecheng, jingfe, zmingy, zhaoyuz,
luqn, xining, lhs, alexbyin, lahl, lingyunw}@amazon.com
jhy0410@tamu.edu
{rulins,menkw23}@uw.edu

Abstract

Large Language Models (LLMs), exemplified by Claude and LLama, have exhibited impressive proficiency in tackling a myriad of Natural Language Processing (NLP) tasks. Yet, in pursuit of the ambitious goal of attaining Artificial General Intelligence (AGI), there remains ample room for enhancing LLM capabilities. Chief among these is the pressing need to bolster long-context comprehension. Numerous real-world scenarios demand LLMs to adeptly reason across extended contexts, such as multi-turn dialogues or agent workflow. Hence, recent advancements have been dedicated to stretching the upper bounds of long-context comprehension, with models like Claude 3 accommodating up to 200k tokens, employing various techniques to achieve this feat. Aligned with this progression, we propose a leaderboard **LongLeader** that seeks to comprehensively assess different long-context comprehension abilities of diverse LLMs and context length extension strategies across meticulously selected benchmarks. Specifically, we aim to address the following questions: 1) Do LLMs genuinely deliver the long-context proficiency they purport? 2) Which benchmarks offer reliable metrics for evaluating long-context comprehension? 3) What technical strategies prove effective in extending the understanding of longer contexts? We streamline the evaluation process for LLMs on the benchmarks, offering open-source access to the benchmarks and maintaining a dedicated website for leaderboards. We will continuously curate new datasets and update models to the leaderboards.

1 Introduction

Long-context Large Language Models (LLMs) refer to those LLMs that can handle very long input or output lengths, by understanding and generating, retrieving and reasoning over long text sequences. As LLMs have stronger capabilities and wider applications (Yang et al., 2024), its long-context capability becomes a vital bottleneck for various use cases (e.g. in-context learning, retrieval augmented generation, agent workflow etc.) (Agarwal et al., 2024; Xu et al., 2023; Weng, 2023).

Prior work regarding long-context LLMs has proposed various training- or inference-stage methods that can extend LLM's contexts (Xiong et al., 2023; Jin et al., 2024), which shows reasonable and improved performance on long-context benchmarks (Bai et al., 2023; Zhang et al., 2024b). However, different benchmarks focus on different perspectives of long context capabilities of LLMs, and different methods report fluctuated benchmark performances, because there is no gold standard for reporting unified benchmark scores regarding various long context capabilities, leading to unclear understanding the effectiveness of various long context extending methods.

To this end, we propose **LongLeader**, a standard and unified long-context LLM benchmark which covers various dimensions, where various LLMs and long-context extending methods can be fairly and comprehensively compared. We also continually pretrained LLMs with various context extending methods, and reported performances of these methods on our unified benchmark, shedding lights on the effectiveness of those methods.

Specifically, although there are many long-context benchmarks proposed and frequently used for evaluating long-context LLMs, the reliability of each datasets is unclear. Also, one benchmark typically focus on sepecific dimensions of long context capability. Meanwhile, metrics and evaluation

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 8734–8750
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

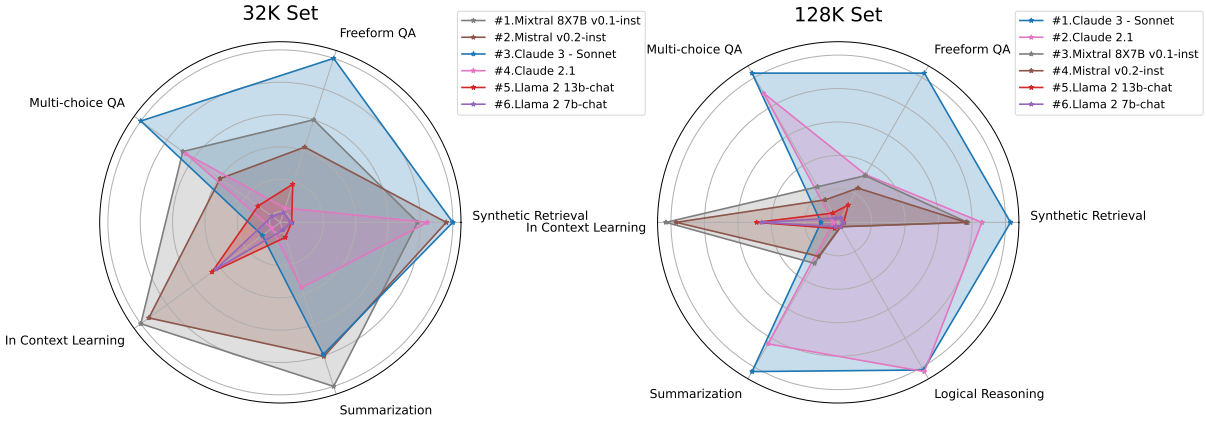

Figure 1: The Leaderboards for both the 32K set and the 128K set. Models in the legends are ranked by their averaged performance.

details are typically not unified for different models, even on the same benchmark, leading to unfair comparison of models, and even over-claimed effectiveness of some methods. To address this, we carefully select some representative datasets to compose a new reliable and comprehensive benchmark, covering 6 dimensions of long-context capabilities (i.e. Multi-choice and Freeform QA, In Context Learning, Summarization, Logical Reasoning and Synthetic Retrieval). To ensure a fair comparison of a model's long-context capability across various context lengths, we define the long context capability for both 0-32k and 0-128k token ranges. We then evaluate LLMs under each setting, with the 128k setting encompassing the 32k subset.

With our proposed benchmark, we re-evaluated several off-the-shelf leading open and closed LLMs, in order to have a fair comparison of those models, assessing whether each of them achieve the long context capability they claimed. For open models, we selected Llama 2 (Touvron et al., 2023) and Mistral&Mixtral (Jiang et al., 2023) etc. as the representative models. For closed models, we chose the Claude family models, because they are well-recognized models with leading performance.

With a pretrained base LLM, it is still unclear with which continual pretraining methods, we can extend the limited context length to our desired context length. Thus, we also continually pretrain Open LLMs with various context-length extension methods, and evaluate them on our proposed benchmarks, which enables us to demystifying which methods can most effectively extend long contexts of LLMs during continual pretraining.

According to the results of off-the-self LLMs and our continually pretrained LLMs on our benchmark, our major findings are summarized as follows:

- Stronger models typically have better long context capabilities, with larger gaps between the best closed and open models on more difficult tasks, like logical reasoning and in-context learning.
- YaRN (Peng et al., 2023) is a generally better choice for continual pretraining to extend context windows, while Amplified Base Frequenc (ABF)-based (Xiong et al., 2023) models enables long-context capability beyond the training window.
- Long context is still challenging for LLMs, especially when it comes to consistent generation and noisy retrieval.

2 Benchmark Selection

To fortify the robustness and ensure the reliability of our analytical processes, we instituted a detailed and systematic selection protocol for assessing existing long-context evaluation benchmarks. A benchmark of high calibre should:

- Reflect the capabilities that are required in handling long-context tasks in real-world applications;
- Effectively differentiate the performance capabilities of various reader models when processing inputs of varying lengths.
- Ensure that the tasks presented incorporate high-quality data to facilitate insightful and substantial evaluations and are balanced—not overly arduous nor unduly simplistic[1].

[1]For example, nearly all models, including GPT-4, have nearly zero performance on the Math.Calc task of InfiniteBench and the Discovery task of LongICLBench.

Benchmark	Data Source		Task Diversity						Length Granularity	
	Synthetic	Realistic	SYR	F-QA	MC-QA	ICL	SUM	LGR	0-32K	>32K
PasskeyRetrieval	✓	✗	✓	✗	✗	✗	✗	✗	✓	✓
NeedleInAHaystack	✓	✗	✓	✗	✗	✗	✗	✗	✓	✓
LongBench	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗
InfiniteBench	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓
LooGLE	✓	✓	✓	✓	✗	✓	✓	✗	✓	✗
RULER	✓	✗	✓	✓	✗	✗	✗	✓	✓	✓
LongEval	✓	✗	✓	✗	✗	✗	✗	✗	✓	✗
LongICLBench	✗	✓	✗	✗	✗	✓	✗	✗	✓	✗
Ours	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparison with Prior Benchmarks.

The first criterion underscores the necessity for language models to possess a range of capabilities to function effectively in real-world long-context applications. Different benchmarks necessitate varied capabilities from these models beyond merely handling long-context tasks. Consequently, our discussion of the models' long-context performance includes a deliberate focus on additional required capabilities specific to each benchmark. This approach enables us to distinctly analyze and identify failures attributable to deficiencies in dimensions of capabilities other than long-context processing. As a result, we meticulously categorize the benchmarks into six distinct sections: Synthetic Retrieval (SYR), Free-form QA (F-QA), Multi-choice QA (MC-QA), In Context Learning (ICL), Summarization (SUM), and Logical Reasoning (LGR).

Synthetic Retrieval (SYR). SYR section includes 5 sub-tasks: SYT-PASSKEY 5/100/1000[2] (Mohtashami and Jaggi, 2024), SYR-RETRIEVE.KV (Zhang et al., 2024b), and SYR-NEEDLE (Liu et al., 2023). These datasets are constructed using synthetic data and evaluate the model's ability of retrieving a specific type of information from a long context.

Free-form Question Answering (F-QA). In this category, we select four datasets: FQA-MUSIQUE (Trivedi et al., 2022), FQA-NARRATIVEQA (Kočiskỳ et al., 2018), FQA-HOTPOTQA (Yang et al., 2018), and FQA-EN.QA (Zhang et al., 2024b). MUSIQUE and HOTPOT-QA are multi-hop QA, which evaluate the model's capability to conduct multihop reasoning based on a long context. NARRATIVEQA contains question-answer pairs that require the model to answer questions about stories by reading entire books or movie scripts. EN.QA, which is a novel-based QA benchmark that requires the model to locate and process information within the novel and perform reasoning through aggregation or filtering to derive answers. These datasets evaluate the capability of the model to locate, aggregate, and reason over the information extracted from the long passages.

Multi-choice QA (MC-QA). MC-QA can be considered as a simplified version of F-QA, which only requires the model to choose a correct answer from the given choices. This category is helpful in distinguishing the capabilities of less capable models. We choose MC-QA-BESTCHOICE (Wang et al., 2024a) and MC-QA-EN.MC (Zhang et al., 2024b) in this category.

In Context Learning (ICL). This category includes 3 tasks in a few-shot form that evaluate the model's in-context learning ability: ICL-DIALOG-RE (Yu et al., 2020), ICL-FEWNERD (Ding et al., 2021), and ICL-TRIVIAQA (Joshi et al., 2017).

Summarization (SUM). This category requires the model to summarize the given long passage. It contains 3 datasets: SUM-QMSUM (Zhong et al., 2021), SUM-GOVREPORT (Huang et al., 2021), and SUM-EN.SUM (Zhang et al., 2024b).

Logical Reasoning (LGR). LGR contains math, code and reasoning domains, which requires the model to be capable of doing logical reasoning based on a long context. We include LGR-CODE.DEBUG (Zhang et al., 2024b) and LGR-MATH.FIND (Zhang et al., 2024b) in this category. LGR-Code.Debug is a multi-choice task requiring the model to find the bug injected to a repository. In LGR-Math.Find, the model receives a array of numbers and is required to locate the largest, the smallest, and the median numbers.

Among the selected tasks, some tasks use real-world context (e.g., En.MC, Code.Debug) which

[2]The number means the length of passkey in digits.

are closer to real-world applications, while other tasks that use curated synthetic contexts (e.g., SYR-Passkey, SYR-Retrieve.KV) are suitable for testing certain capabilities of long-context LLMs without entangling complex capabilities that are unrelated to the long-context ability. More details can be found in Appendix B.

Additionally, to adequately address varying depths of context comprehension and aggregate the benchmarks for easy interpretation, we divide the benchmarks into two primary sets based on the length of contexts they are designed to evaluate: *32K Set* and *128K Set*, which evaluate the model's ability to handle context length up to 32k tokens and 128k tokens, respectively. The segmentation of the datasets into the two sets, along with the specifics of each dataset, is detailed in Table 3. Our structured approach ensures a holistic and detailed evaluation of different context lengths, serving as the foundation of our benchmark. Note that we define the long context capability of the two sets as 0-32k and 0-128k token ranges, with the 128k setting encompassing the 32k subset. A comparison of our benchmark with prior works is shown in Table 1.

3 Leaderboards

In this section, we present the leaderboards for our selected benchmarks. We feature two primary leaderboards, comparing 10 widely-used and popular public models across the two sets of benchmarks, each with context lengths of 32k/128k. Additionally, we include an auxiliary leaderboard to assess various context extension techniques using the same continuing pretraining recipe.

3.1 Settings

Counting Tokenizer Selection: Different tokenizers may yield varying token counts for the same text. To ensure a fair comparison across all models, we adopt the Llama 2 (Touvron et al., 2023) tokenizer as the standard for token counting. Our choice is primarily based on its relatively lower compression rate. This ensures that the token length counted by Llama 2 does not exceed that counted by many other tokenizers. This approach mitigates potential discrepancies that could impact the fairness of the leaderboard comparison.

Model Selections We carefully select 10 models for evaluation based on three principles: 1) covering various context window lengths, 2) including widely used open and closed models, and 3) representing a range of model sizes. For the closed models, we select Claude 2.1/3-Sonnet, primarily because of their demonstrated performance and long-context capabilities. For the open models, we choose Mistral/Mixtral (Jiang et al., 2023, 2024) due to their outstanding performance compared to many similar size LLMs. We select the Llama-2 series (7B, 13B) (Touvron et al., 2023) because it is one of the most widely used model family. More Details about the context window and the version of the selected models are shown in Appendix B Table 4.

Context Extension Recipe: We study position extension techniques based on RoPE (Su et al., 2024), which is one of the most widely adopted position embedding methods. Recently released LLMs predominantly use RoPE for position embedding. More technical details about RoPE can be found in Appendix A. To assess the efficacy of cutting-edge long-context extension techniques with our selected 32k benchmark, we continue pretraining Llama 2 models using Linear Position Interpolation (PoI) (Chen et al., 2023b), YaRN (Peng et al., 2023), and Amplified Base Frequency (ABF) (Xiong et al., 2023). These techniques were chosen based on the following criteria: (a) They are among the most widely adopted methods for extending context window lengths, forming the foundation of many long-context LLMs; (b) They require minimal modifications and can be easily applied to any RoPE-based models.

We opt for Llama 2 7b/13b as the base models due to their fixed pretrained context window of 4k (in contrast to Mistral/Mxitral (Jiang et al., 2023, 2024) models, which have already undergone continued pretraining for context window extension). By extending the context window to 32k, we evaluate its performance against our 32k benchmark. We select the 5B tokens curated by (Fu et al., 2024) for training due to their high data quality and manageable training overheads. In order to enable the instruction following ability of the models for conveniently evaluation, we utilize the LongAlpaca (Chen et al., 2024) dataset with its 12k samples, chosen for its data quality and a context window size of 32k, aligning well with our evaluation parameters.

We choose the RoPE Scaling Factor as 8.0 for both the PoI and YaRN continued pretraining, since we extend the models from 4k to 32k with 8× extensions. As for ABF, we adopt the default am-

plified factor as 50.0, which means, we amplify the RoPE theta from 10000.0 to 500000.0. More detailed settings can be found in Table 8, 10, 9, 11.

Evaluation Protocols: To ensure a fair comparison, we standardize the evaluation metrics for all datasets, as shown in Appendix B Table 3. We adopt `Exact Match` to calculate the correctness rate with each test sample for the SYR-PASSKEY5/100/1000, SYR-RETRIEVE.KV, MC-QA (BESTCHOICE and EN.MC), LGR (CODE.DEBUG and MATH.FIND) datasets. For all the F-QA and ICL tasks, we adopt the `F1 score` of the generated tokens in comparison to the golden answers. For all the SUM tasks, we adopt the ROUGE-L-Sum metric (Lin, 2004) as in the previous work. Specially, for the SYR-NEEDLE task, we also adopt a third LLM as a judge to score the answers, and the score is normalized from (1,10) to (0, 1). Clause 3 Sonnet is utilized as the judge.

As for the models, we set the temperature as 0 with greedy decoding for all models in the evaluations.

3.2 Main Results

To calculate the leaderboard rankings, we average all task scores for each type across both the 32k Set and the 128k Set. For the overall average score of each model, we use the average across task types to prevent over-weighting any type with more sub-tasks. Furthermore, please note that for the 128k benchmark (able 2), we directly average the scores from the 32k (Appendix C Table 5) set and the 32-128k (Appendix C Table 6) set to avoid over-weighting the 32k benchmarks, which contain more sub-tasks. As shown in Figure 1, Figure 2 we can conclude the following observations:

- Claude 3-Sonnet outperforms the competition in both the 32k and 128k benchmarks, demonstrating consistently excellent performance across various task dimensions.
- LLama2 7b/13b consistently rank lower, primarily because of their relatively short context length.
- Surprisingly, Mistral and Mixtral perform excellently in the 32k set, surpassing Claude 2.1/3-Sonnet in this context length range.
- Claude 2.1/3-Sonnet can handle longer contexts than the Mistral/Mixtral models, and their performance in the 128k set is superior.
- Overall, YaRN-based models perform better than the PoI and ABF models consistently across different model sizes. However, their performance varies when it comes to different long-context abilities.

4 Analysis and Insights

To extend the context window by continue pre-training, YaRN is better. From Table 7, the overall performance of YaRN is better than ABF and PoI. It's mainly because they interpolate lower dimensions and higher dimensions with different strategies:

- PoI uniformly interpolates all dimensions in a LLM using RoPE. However, RoPE's design implies that lower dimensions do not require interpolation and training them to accommodate manipulations is challenging as noted by (Peng et al., 2023). Additionally, the scale factor for PoI typically matches the extension factor, leading to the utilization of some under-trained positions during the extension of the context window. For instance, in Llama-2, which has a pretraining context window of 4096, for a input with 4096 tokens, the relative position 4095 appears only once, while position 2047 appears 2048 times, and position 0 appears 4096 times. Consequently, larger positions like 4095 are significantly under-trained compared to smaller positions such as 2047 and 0. Supporting this observation, the performance disparity between PoI and YaRN becomes more pronounced on tasks with many instances close to 32k, such as ICL-Dialog-RE and SYR-Passkey, but diminishes on tasks with instances around 16k, like F-QA-Musique and F-QA-HotpotQA.
- With ABF, higher dimensions are more interpolated while lower dimensions receive less interpolation. Theoretically, ABF should surpass PoI as it has no excessive interpolation of lower dimensions (Xiong et al., 2023). Contrary to expectations and previous findings such as those by (Xiong et al., 2023), ABF is the worst one in our benchmark. We hypothesize that this discrepancy arises because, unlike PoI, ABF typically employs a much larger theta scaling factor relative to its extension scale in common practice, as was the case in our experiments. The much larger scale factor will need more training to fit the much closer new positions compared to PoI's smaller scale factor. It should be noted

Task Type	Claude 3	Claude 2.1	Llama 2 13b-chat	Llama 2 7b-chat	Mixtral	Mistral v0.2-inst
SYR	98.28	82.76	7.59	7.32	74.96	74.20
F-QA	34.25	17.11	12.00	9.81	16.95	14.85
MC-QA	56.20	48.64	3.43	1.57	13.35	8.42
ICL	12.31	9.09	29.23	27.89	53.13	50.44
SUM	24.44	22.00	11.83	11.68	14.91	14.29
LGR	20.78	21.01	N/A	N/A	N/A	N/A
AVG	41.04	33.44	10.68	9.71	28.88	27.03

Table 2: Test Results for the Whole 128K Benchmark (%)

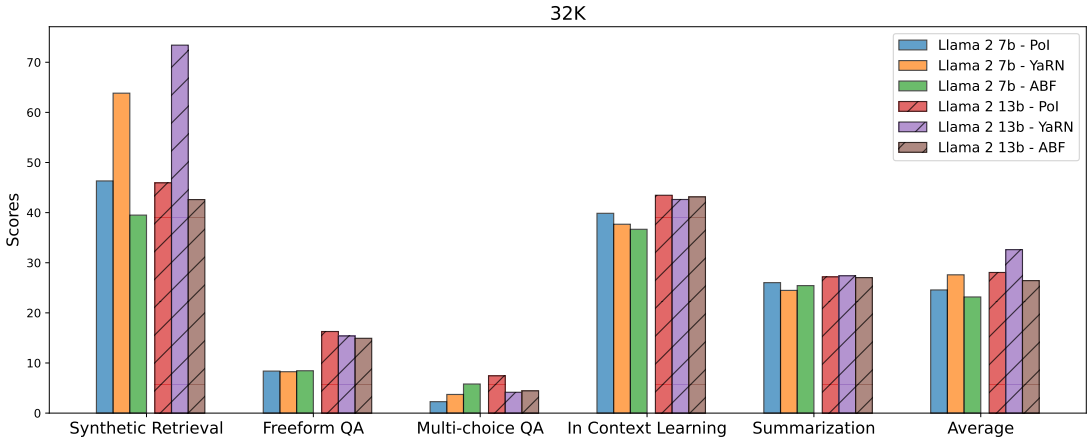

Figure 2: Leaderboard for the Position Extension Techniques on the 32K Set Benchmark.

that (Xiong et al., 2023) utilized considerably more training data (400B) and computational resources than our experiments. While we speculate that ABF's increased data and computation demands might explain the divergence in results, it is important to acknowledge that this is merely a conjecture without concrete evidence. Thus, our findings suggest that under the same conditions of 5B tokens, ABF may not perform as expected due to its potential data-intensive nature.

- YaRN does not interpolate in lower dimensions. For higher dimensions it adapts the strategy of ABF. In a short range, relative positions are heavily relied on those lower dimensions, which is crucial to next token prediction. Hence YaRN can better modeling the position relations in a short range. With un-interpolated lower dimensions, Yarn can directly leverage those well trained short-range relative positions from the pretraining stage. Meanwhile, different from PoI and ABF, YaRN incorporates a scaling factor in the softmax function of self-attention. This adjustment helps the attention be more focused on important tokens. This potentially contributing to YaRN's advantage over PoI and ABF.

A strong base model means strong long context abilities. Long context capability is nothing different from other perspectives of LLMs. It's mainly from the base model. We can clearly see that, with the same fine-tuning data and same extension methods, larger models perform much better than small models. Most long context tasks have a bunch of noisy or useless information. Generally, larger models are more capable of focusing on and extracting useful information from the input texts. We argue that this is one of the major reasons of larger models' superiority on long context tasks. Also, with similar size, stronger base models implicitly have stronger long context abilities. To further illustrate this, we conduct more experiments for Llama-2-7b, Llama-2-70b and Llama-3-8b on the Needle-in-a-Haystack task. We use SelfExtend (Jin et al., 2024) to extend the three models' context window. We suppose SelfExtend, as a fine-tuning free method, will not change models' abilities. From Appendix E Table 12. We can see the ranking of the three models are nearly the same as the ranking of their performance on standard short tasks (AI@Meta, 2024). We also suppose that, larger models have more obvious superiority to small models on longer context tasks than on

similar tasks but with short contexts.

In a short summary, all extension methods primarily handle the out-of-distribution problem in long context extension. This means with the same extension method and the same data, the effectiveness of the long-context model relies on the quality of the base model. Good base models have strong abilities to ignore noisy information, which is required by most long-context tasks.

Consistent long context generation is challenging. The original passkey retrieval task (Mohtashami and Jaggi, 2024) uses a 5-digit passkey. It can be perfectly (i.e. 100% accuracy) solved by many existing long context handling methods (Peng et al., 2023; Xiong et al., 2023; Chen et al., 2024; Jin et al., 2024; Xiao et al., 2024). But it becomes different with the introduction of longer passkeys. Our findings indicate a decline in performance as the length of the passkey increases, as in Appendix C Table 5 and 6. While all models demonstrate proficiency with shorter passkeys, their effectiveness varies significantly with longer sequences.

Super long passkey challenges LLMs' abilities to precisely memorize useful information and generate consistent prediction according to it. Compared to standard short context tasks, usually, long context tasks have lower information density. The model just needs a small part of the context to make predictions. If the required information is short, it's easy for a model to memorize the information. As the required information lengthens, the task of accurately memorizing and regenerating every detail becomes increasingly complex. Moreover, the prediction of longer passkeys demands a higher tolerance for error accumulation during inference.

Supporting this observation, nearly all failure cases still successfully identify the location of the passkey, but they cannot precisely repeat the whole passkey sequence, such as missing parts of the passkey or wrongly repeating several digits. This inconsistency underscores the challenges faced by LLMs in generation of long sequences.

With high noise level, retrieval tasks can be difficult. Retrieval is one of the most fundamental tasks in machine learning. Typically, these tasks do not require complex reasoning abilities; however, they can become significantly challenging with increased noise levels. All models generally perform well within their pretraining context windows on passkey retrieval tasks. Conversely, their performance tends to diverge during key-value (KV) retrieval tasks. Both tasks require the large language model (LLM) to retrieve a simple message from the context according to a specified 'key'. The primary distinction between the two is that KV retrieval involves a noisier context compared to passkey retrieval. Specifically, the context in KV retrieval comprises other 'key-message' pairs, whereas the context in passkey retrieval consistently repeats a single, irrelevant sentence multiple times, which bears no relation to the target 'key-message' pair. This results in differents failure patterns of the two tasks: KV retrieval failures often occur when the model is distracted by competing pairs and extracts the incorrect pair, whereas passkey retrieval failures typically involve the model's inability to generate the complete sequence as described earlier. Similar findings are reported by RULER (Hsieh et al., 2024), which designed several 'Needle-in-a-Haystack' variants by adjusting noise levels. In these studies, LLMs consistently exhibited poorer performance on variants with higher noise levels.

Nevertheless, simple retrieval tasks like passkey retrieval remain valuable as they serve as touchstone for assessing whether a model can access all information in an input sequence. Existing research (Jin et al., 2024; Arora et al., 2024, 2023) indicates that some linear-time models or attention mechanism, despite abilities comparable to vanilla transformers, have a limited receptive field to the input sequence, leading to their failure in these simple retrieval tasks. We plan to test such models in the near future.

ABF-based models can somewhat work beyond their pre-training windows. Despite the context window being only 8k for Mixtral and 32k for Mistral, these models effectively handle information retrieval tasks involving up to 128k contexts. What happens here is similar to what is observed in LongLora (Chen et al., 2024): after interpolation, although continuously trained on short sequences, LLMs are able to generalize to longer sequences. More details about the connection between this phenomenon and existing works are in Appendix F.

5 Related Work

Long Context Benchmarks. Our work is closely related to other works on benchmarking long-context language models. LongEval (Dacheng Li*, 2023) is one of the pioneer work which includes various long context retrieval tasks. Long-

Bench (Bai et al., 2023) contains various long-context tasks in a bilingual setting. It includes variable task types such as question answering, coding, summarization, and others. InfiniteBench (Zhang et al., 2024b) is similar to LongBench, but its tasks have a greater than 100K token length, while most data in LongBench is less than 20K tokens. Some tasks in InfiniteBench are so difficult that nearly no models can handle them. Ada-LEval (Wang et al., 2024b) introduces novel segment sorting and many-choice selection tasks. It has a more fine-grained length distribution ranging from 1K to 128K tokens. LongICLBench (Li et al., 2024a) targets challenging the in-context learning abilities of large language models (LLMs). For each task, it constructs the data in rounds. A round means a complete set of all candidate labels. Each task includes settings of 1 to 5 rounds, varying the context to investigate the influence of the number of examples in in-context learning. RULER (Hsieh et al., 2024) is designed to challenge the true context window length of long-context LLMs. To control the task complexity and data length, it is composed entirely of synthetic data such as variants of needle-in-a-haystack by adjusting the number of needles and the contents of the contexts. LooGLE (Li et al., 2023) tries to test LLMs' abilities to extract and understand long-range dependencies over the entire sequence. The data in LooGLE is mainly around 32K tokens.

Context Window Extension Methods. Several methods extend the context window for LLMs: *Retrieval-Based Approaches:* These use an external memory module to store past context and fetch related documents during inference, necessitating modifications to LLM architectures. Examples include Activation Beacon (Zhang et al., 2024a) and Landmark Attention (Mohtashami and Jaggi, 2024). *Fine-Tuning Based Approaches:* One line of these methods interpolates long context positions into the original LLM context window, optimizing pre-trained models with large base values for position embeddings. Notable implementations are CodeLLaMA (Roziere et al., 2023) and LlamaLong (Xiong et al., 2023). LongLoRA (Chen et al., 2024) and CLEX (Chen et al., 2023a) reduce GPU resource demands by fine-tuning on sequences shorter than the target length. Another line uses short sequences to mimic long sequences during fine-tuning such as PoSE (Zhu et al., 2023). *Attention-Based Methods:* Beyond position embedding interpolation, these methods adapt attention mechanisms to manage input context without requiring fine-tuning. Sliding Window Attention limits attention to nearby tokens but fails to retain distant token information (Jiang et al., 2023). StreamLM (Xiao et al., 2023) and LLM-Infinite (Han et al., 2023) apply similar strategies, focusing on head and neighboring tokens while masking intermediates. SelfExtend (Jin et al., 2024) maintains all input tokens but only retains accurate positional information for tokens within a close range, simplifying distant positions to those seen during pre-training. *KV Cache Utilization:* Techniques such as Heavy-Hitter Oracle (Zhang et al., 2024c), Snap-KV (Li et al., 2024b) and InfLLM (Xiao et al., 2024) develop KV cache eviction policies that optimize token retention based on attention scores or token window relevance during generation, thereby efficiently managing memory constraints.

6 Conclusion

In this work, we propose a comprehensive benchmark for evaluating long-context Large Language Models, addressing the lack of a unified standard for consistent performance comparison. Our benchmark covers various dimensions and context lengths, enabling a fair comparison of different LLMs and context-extending methods. This approach provides clearer insights into the effectiveness of these models and methods.

7 Limitations

To evaluate the proposed six types of long-context abilities across various context lengths, we conducted a comprehensive literature review and carefully selected off-the-shelf datasets to cover different lengths and abilities. However, we could not identify suitable logical reasoning tasks for the 32k length, leaving this area open for future research. Furthermore, it is important to acknowledge that the format of the prompt can influence the outcomes of the evaluation. To maintain consistency, we implemented a standardized prompt format across all models in our assessment. This approach, however, may introduce a degree of bias, as different models may perform optimally with varying prompt formats. Addressing this potential unfairness and exploring a more equitable setup will be the focus of future research efforts.

References

Rishabh Agarwal, Avi Singh, Lei M Zhang, Bernd Bohnet, Stephanie Chan, Ankesh Anand, Zaheer Abbas, Azade Nova, John D Co-Reyes, Eric Chu, et al. 2024. Many-shot in-context learning. *arXiv preprint arXiv:2404.11018*.

AI@Meta. 2024. Llama 3 model card.

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Ré. 2023. Zoology: Measuring and improving recall in efficient language models. *arXiv:2312.04927*.

Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, Dylan Zinsley, James Zou, Atri Rudra, and Christopher Ré. 2024. Simple linear attention language models balance the recall-throughput tradeoff. *arXiv:2402.18668*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *Preprint*, arXiv:2308.14508.

Guanzheng Chen, Xin Li, Zaiqiao Meng, Shangsong Liang, and Lidong Bing. 2023a. Clex: Continuous length extrapolation for large language models. *arXiv preprint arXiv:2310.16450*.

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

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2024. Longlora: Efficient fine-tuning of long-context large language models. *Preprint*, arXiv:2309.12307.

Anze Xie Ying Sheng Lianmin Zheng Joseph E. Gonzalez Ion Stoica Xuezhe Ma Hao Zhang Dacheng Li*, Rulin Shao*. 2023. How long can open-source llms truly promise on context length?

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Hai-Tao Zheng, and Zhiyuan Liu. 2021. Few-nerd: A few-shot named entity recognition dataset. *arXiv preprint arXiv:2105.07464*.

Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. 2024. Data engineering for scaling language models to 128k context. *Preprint*, arXiv:2402.10171.

Chi Han, Qifan Wang, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. 2023. Lm-infinite: Simple on-the-fly length generalization for large language models. *arXiv preprint arXiv:2308.16137*.

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

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. *arXiv preprint arXiv:2104.02112*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

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. *arXiv preprint arXiv:2401.01325*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

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

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024a. Long-context llms struggle with long in-context learning. *Preprint*, arXiv:2404.02060.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024b. Snapkv: Llm knows what you are looking for before generation. *arXiv preprint arXiv:2404.14469*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

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

Amirkeivan Mohtashami and Martin Jaggi. 2024. Random-access infinite context length for transformers. *Advances in Neural Information Processing Systems*, 36.

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

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

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

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

Lilian Weng. 2023. Llm-powered autonomous agents. *lilianweng.github.io*.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, Song Han, and Maosong Sun. 2024. Infllm: Unveiling the intrinsic capacity of llms for understanding extremely long sequences with training-free memory. *arXiv preprint arXiv:2402.04617*.

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

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, Madian Khabsa, Han Fang, Yashar Mehdad, Sharan Narang, Kshitiz Malik, Angela Fan, Shruti Bhosale, Sergey Edunov, Mike Lewis, Sinong Wang, and Hao Ma. 2023. Effective long-context scaling of foundation models. *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*.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. 2024. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–32.

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

Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. 2020. Dialogue-based relation extraction. *arXiv preprint arXiv:2004.08056*.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. 2024a. Soaring from 4k to 400k: Extending llm's context with activation beacon. *Preprint*, arXiv:2401.03462.

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. 2024b. ∞bench: Extending long context evaluation beyond 100k tokens. *Preprint*, arXiv:2402.13718.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. 2024c. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, et al. 2021. Qmsum: A new benchmark for query-based multi-domain meeting summarization. *arXiv preprint arXiv:2104.05938*.

Dawei Zhu, Nan Yang, Liang Wang, Yifan Song, Wenhao Wu, Furu Wei, and Sujian Li. 2023. Pose: Efficient context window extension of llms via positional skip-wise training. *arXiv preprint arXiv:2309.10400*.

A Preliminary

RoPE. Modern LLMs typically use the relative positions of input tokens. One method to encode this information is through Rotary Position Embedding (RoPE). Let's consider a sequence of tokens represented as $w_1, w_2, \cdots, w_L$, and their corresponding embeddings are denoted as $\mathbf{x}_1, \cdots, \mathbf{x}_L \in \mathbb{R}^{|D|}$, where $|D|$ is the dimension of the embedding. The basic idea of RoPE is to incorporate the positional information into the query $\mathbf{q}$ and the key vectors $\mathbf{k}$, respectively. This integration ensures that their inner product $\mathbf{q}^T\mathbf{k}$ will contain the relative positional embedding information inherently. To achieve this, RoPE employs the following vector transformations: $\mathrm{q}_m = f_q(\mathbf{x}_m, m) \in \mathbb{R}^{|L|}$, $\mathbf{k}_n = f_k(\mathbf{x}_n, n) \in \mathbb{R}^{|L|}$, where $|L|$ is the hidden dimension of per head. The functions f_q and f_k responsible for injecting positional information, are defined as $f_q(\mathbf{x}_m, m) = W_q\mathbf{x}_m e^{im\theta}$, $f_k(\mathbf{x}_n, n) = W_k\mathbf{x}_n e^{in\theta}$, where $\theta_d = b^{-2d/|D|}$, $b = 10000$ and projectors $W_q, W_k : \mathbb{R}^{|D|} \rightarrow \mathbb{R}^{|L|}$. RoPE keeps the real part of the inner product $\mathbf{q}^T\mathbf{k}$, which is $Re(\mathbf{q}^*\mathbf{k})$. This operation ensures that the dot product of the query and key vectors depends entirely on the relative distance between the tokens, represented by $m - n$ of the tokens as follows: $\langle f_q(\mathbf{x}_m, m), f_k(\mathbf{x}_n, n)\rangle_{\mathbb{R}} = Re(\langle f_q(\mathbf{x}_m, m), f_k(\mathbf{x}_n, n)\rangle_{\mathbb{C}}) = Re(\mathbf{x}_m^* W_q^* W_k \mathbf{x}_n e^{i\theta(m-n)}) = g(\mathbf{x}_m, \mathbf{x}_n, m - n)$,where $g(\cdot)$ is an abstract mapping function.

B Details of the Selected Benchmarks and Models

Table 3 shows the statistics of the selected benchmarks and Table 4 shows the details of the selected models for evaluation.

Task Type	Dataset	Length	Average Length # Tokens	# Example	Metric
SYR	Passkey 5 - 32k	32K	16986.50	160	EM (Exact Match)
	Passkey 100 - 32k	32K	16985.50	160	EM (Exact Match)
	Passkey 1000 - 32k	32K	17985.75	80	EM (Exact Match)
	Needle - 32k	32K	16801.00	176	Claude Rating
	Retrieve.KV - 32k	32K	31664.82	90	EM (Exact Match)
	Passkey 5 - 128k	128K	64988.19	640	EM (Exact Match)
	Passkey 100 - 128k	128K	64987.19	640	EM (Exact Match)
	Passkey 1000 - 128k	128K	65986.63	320	EM (Exact Match)
	Needle - 128k	128K	64801.00	704	Claude Rating
	Retrieve.KV - 128k	128K	126441.36	360	EM (Exact Match)
F-QA	Musique	32K	18556.32	200	F1
	NarrativeQA	32K	36083.33	200	F1
	HotpotQA	32K	15330.78	200	F1
	En.QA	128K	99337.86	92	F1
MC-QA	BestChoice	32K	10320.51	4200	EM (Exact Match)
	En.MC	128K	102713.44	63	EM (Exact Match)
ICL	Dialog-RE	32K	20270.88	354	F1
	FewNerd	32K	14191.22	1500	F1
	TriviaQA	32K	14099.07	200	F1
SUM	QMSUM	32K	15981.50	200	Rouge-L
	GovReport	32K	12281.15	200	Rouge-L
	En.Sum	128K	96854.84	37	Rouge-L
LGR	Code.Debug	128K	113426.43	155	EM (Exact Match)
	Math.Find	128K	117918.33	350	EM (Exact Match)

Table 3: Benchmark Statistics

Models	Version	Context Window
Claude	*claude-v2.1*	200K
	claude-3-sonnet-v1.0	200K
LLama 2 (Touvron et al., 2023)	*Llama-2-7b-chat*	4K
	Llama-2-13b-chat	4k
Mistral (Jiang et al., 2023)	*Mistral-7B-Instruct-v0.2*	32K
	Mixtral-8X7B-Instruct-v0.1	8k

Table 4: Model Selections

C Details of the Evaluation Results

Here are the details results for the evaluations. Table 5 shows the results of the 32K Set benchmark and Table 2 showcases the 128K Set benchmark. Note that the Table 6 shows the details of the benchmarks that are ranged from 32K-128K, and it does not include the 32K benchmark. Table 7 shows the results of different position extension techniques on the 32K Set benchmark.

Task Type	Task	Claude 3	Claude 2.1	Llama 2 13b-chat	Llama 2 7b-chat	Mixtral	Mistral v0.2-inst
SYR	Passkey 5	97.50	96.25	12.50	11.25	100	100
	Passkey 100	98.75	100	12.50	12.50	100	100
	Passkey 1000	100	100	8.75	8.75	100	100
	Needle	99.89	92.44	20.97	20.00	99.49	98.30
	Retrieve.KV	100	37.78	0	0	0	80
	AVG	99.23	85.29	10.94	10.50	79.90	95.66
F-QA	Musique	39.98	8.88	13.53	8.80	27.35	18.64
	NarrativeQA	28.81	19.75	19.04	16.90	26.47	26.61
	HotpotQA	61.19	32.44	39.44	33.17	47.87	43.84
	AVG	43.33	20.36	24.00	19.62	33.90	29.70
MC-QA	BestChoice	37.80	25.84	6.86	3.14	26.70	16.84
	AVG	37.80	25.84	6.86	3.14	26.70	16.84
ICL	Dialog - RE	2.14	0	N/A	N/A	29.82	19.31
	Few Nerd	1.18	13.06	N/A	N/A	38.84	46.21
	TriviaQA	33.62	14.22	87.69	83.67	90.74	85.81
	AVG	12.31	9.09	29.23	27.89	53.13	50.44
SUM	QMSum	22.79	21.44	20.97	20.55	26.05	24.63
	GovReport	34.21	30.03	26.35	26.17	33.59	32.52
	AVG	28.50	25.74	23.66	23.36	29.82	28.58
AVG	-	44.23	33.26	18.94	16.90	44.69	44.24

Table 5: Test Results for the 32K Benchmark (%)

Task Type	Task	Claude 3	Claude 2.1	Llama 2 13b-chat	Llama 2 7b-chat	Mixtral	Mistral v0.2-inst
SYR	Passkey 5	98.28	98.75	3.13	2.81	100	98.44
	Passkey 100	99.69	100	3.13	3.13	96.41	65.63
	Passkey 1000	92.50	100	2.19	2.19	79.69	46.88
	Needle	96.76	94.89	12.74	12.50	73.98	52.68
	Retrieve.KV	99.44	7.50	0	0	0	0
	AVG	97.33	80.23	4.24	4.13	70.02	52.73
F-QA	En.QA	25.17	13.86	N/A	N/A	N/A	N/A
	AVG	25.17	13.86	N/A	N/A	N/A	N/A
MC-QA	En.MC	74.60	71.43	N/A	N/A	N/A	N/A
	AVG	74.60	71.43	N/A	N/A	N/A	N/A
SUM	En.Sum	19.56	18.26	N/A	N/A	N/A	N/A
	AVG	19.56	18.26	N/A	N/A	N/A	N/A
LGR	Code.Debug	13.55	2.58	N/A	N/A	N/A	N/A
	Math.Find	28.00	39.43	N/A	N/A	N/A	N/A
	AVG	20.78	21.01	N/A	N/A	N/A	N/A
AVG	-	47.49	40.96	0.85	0.83	14.00	10.55

Table 6: Test Results for the 128K Benchmark (%, NOT including the 32K benchmark here)

Task Type	Task	Llama 2 7b PoI	Llama 2 7b YaRN	Llama 2 7b ABF	Llama 2 13b PoI	Llama 2 13b YaRN	Llama 2 13b ABF
SYR	Passkey 5	99.38	100	100	97.50	100	85.63
	Passkey 100	48.13	92.50	62.50	32.50	91.88	27.50
	Passkey 1000	0	55.00	0	1.25	81.25	0
	Needle	84.09	71.65	35.00	98.47	93.98	99.88
	Retrieve.KV	0	0	0	0	0	0
	AVG	46.32	63.83	39.50	45.94	73.42	42.60
F-QA	Musique	8.27	7.84	8.17	12.78	12.41	12.36
	NarrativeQA	6.00	6.00	5.71	14.38	15.58	15.04
	HotpotQA	10.86	10.95	11.48	21.67	18.19	17.38
	AVG	8.38	8.26	8.45	16.28	15.39	14.93
MC-QA	BestChoice	2.28	3.72	5.80	7.44	4.16	4.44
	AVG	2.28	3.72	5.80	7.44	4.16	4.44
ICL	Dialog - RE	6.78	7.20	5.72	11.46	12.16	14.10
	Few Nerd	33.93	34.49	25.91	36.85	38.36	34.68
	TriviaQA	78.88	71.34	78.37	82.09	77.35	80.63
	AVG	39.86	37.68	36.67	43.47	42.62	43.14
SUM	QMSum	22.90	21.92	22.24	24.30	23.99	24.51
	GovReport	29.17	27.03	28.64	30.05	30.80	29.55
	AVG	26.03	24.48	25.44	27.18	27.40	27.03
AVG	-	24.57	27.59	23.17	28.06	32.61	26.43

Table 7: Test Results for the 32K Benchmark w/ Different Position Extension Techniques

D Details of Experimental Settings for the Position Extensions

Table 8 and Table 9 shows the hyper-parameter settings and the training cost for the continue-pretraining with the three (PoI, YaRN, and ABF) positions extension strategies on the 5B tokens (Fu et al., 2024). Table 10 and Table 11 showcase the hyper-parameter settings and the training cost for the long-context instruction fine-tuning with LongAlpaca (Chen et al., 2024) dataset.

Models	Scaling Factor	Learning Rate	Warm-up Ratio	Max Epochs	Max Sequence Length	Weight Decay	Optimizer
LLama 2 7b - PoI	8	4 million	2e-5	1	32K	0.1	Adam
LLama 2 7b - YaRN	8	4 million	2e-5	1	32K	0.1	Adam
LLama 2 7b - ABF	50	4 million	2e-5	1	32K	0.1	Adam
LLama 2 13b - PoI	8	4 million	2e-5	1	32K	0.1	Adam
LLama 2 13b - YaRN	8	4 million	2e-5	1	32K	0.1	Adam
LLama 2 13b - ABF	50	4 million	2e-5	1	32K	0.1	Adam

Table 8: Continue Pretraining Hyperparameters

Models	GPUs	Parallel Settings	Accelerator	Precision	Training Time
LLama 2 7b - PoI	64 × NVIDIA A100 80G	TP =8, PP=1, DP=8	DeepSpeed (ZeRO Stage 1)	bf16	1.5 days
LLama 2 7b - YaRN	64 × NVIDIA A100 80G	TP =8, PP=1, DP=8	DeepSpeed (ZeRO Stage 1)	bf16	1.5 days
LLama 2 7b - ABF	64 × NVIDIA A100 80G	TP =8, PP=1, DP=8	DeepSpeed (ZeRO Stage 1)	bf16	1.5 days
LLama 2 13b - PoI	64 × NVIDIA A100 80G	TP =8, PP=2, DP=8	DeepSpeed (ZeRO Stage 1)	bf16	2.9 days
LLama 2 13b - YaRN	64 × NVIDIA A100 80G	TP =8, PP=2, DP=8	DeepSpeed (ZeRO Stage 1)	bf16	2.9 days
LLama 2 13b - ABF	64 × NVIDIA A100 80G	TP =8, PP=2, DP=8	DeepSpeed (ZeRO Stage 1)	bf16	2.9 days

Table 9: Continue Pretraining Cost (TP: Tensor Parallelism, PP: Pipeline Parallelism, DP: Data Parallelism)

Models	Batch Size	Learning Rate	Warm-up Steps	Max Epochs	Weight Decay	Optimizer
LLama 2 7b - PoI	64	2e-5	20	5	0.0	Adam
LLama 2 7b - YaRN	64	2e-5	20	5	0.0	Adam
LLama 2 7b - ABF	64	2e-5	20	5	0.0	Adam
LLama 2 13b - PoI	64	2e-5	20	5	0.0	Adam
LLama 2 13b - YaRN	64	2e-5	20	5	0.0	Adam
LLama 2 13b - ABF	64	2e-5	20	5	0.0	Adam

Table 10: Long-context Instruction Fine-tuning Hyperparameters

Models	GPUs	Accelerator	Precision	Training Time
LLama 2 7b - PoI	8 × NVIDIA A100 80G	DeepSpeed (ZeRO Stage 2)	bf16	~3.0 hour / epoch
LLama 2 7b - YaRN	8 × NVIDIA A100 80G	DeepSpeed (ZeRO Stage 2)	bf16	~3.0 hour / epoch
LLama 2 7b - ABF	8 × NVIDIA A100 80G	DeepSpeed (ZeRO Stage 2)	bf16	~3.0 hour / epoch
LLama 2 13b - PoI	8 × NVIDIA A100 80G	DeepSpeed (ZeRO Stage 2)	bf16	~5.2 hour / epoch
LLama 2 13b - YaRN	8 × NVIDIA A100 80G	DeepSpeed (ZeRO Stage 2)	bf16	~5.2 hour / epoch
LLama 2 13b - ABF	8 × NVIDIA A100 80G	DeepSpeed (ZeRO Stage 2)	bf16	~5.2 hour / epoch

Table 11: Long-context Instruction Fine-tuning Cost

E More Experiment Results

Task	Llama-2-7b-chat	Llama-2-70b-chat	Llama-3-8b-Inst
Needle @32k	0.685	0.940	0.990
Needle @16k	0.940	0.993	0.995

Table 12: For Llama-2 family, 32k is 8× extension and 16k is 4× extension. For Llama-3, 32k is 4× extension and 16k is 2× extension. To make the comparison fairer, both 32k and 16k experiments are conducted for Llama-3-8b-inst.

F More Explanation for Analysis

While we have no knowledge about training details of Mistral and Mistral, their abilities of generalizing beyond their training window can be explained by the reconstruction of precise neighbor position information during fine-tuning.

The term "neighbor" refers to those tokens near to the tokens being generated. This stems from SelfExtend (Jin et al., 2024) and LongLoRA (Chen et al., 2024). Both papers emphasize the importance of precise positions for neighbor tokens, while distant token positions can be less precise. For example, in a 128k sequence, the nearest 8k tokens to the next generated tokens can be treated as the "neighbors", and their positions are most important.

To be more specific, the base models of Mixtral and Mistral have an 8k context window, with a RoPE theta of 10,000. Mixtral and Mistral employ ABF and significantly increase the RoPE theta from 10,000 to 500,000, which is 50 times larger. After fine-tuning on short contexts (32k for Mistral and 8k for Mixtral), for a 128k input, these models have reconstructed precise neighbor position information (the nearest 32k tokens for Mistral and 8k tokens for Mixtral), while distant positions (32k&8k to 128k) are just interpolated by ABF and not that precise. What happens here is very similar to LongLoRA. With PoI, LongLoRA uses a local attention block of 8k to fine-tune a model from 4k to 32k. Essentially, using 8k local attention block is nearly equivalent to directly training the model on 8k sequences. In this way, LongLoRA reconstructs the neighbor positions (8k), and it demonstrates pretty good performance.