Effective Long-Context Scaling of Foundation Models

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[*], Hao Ma[*]

Meta

Abstract

We present a series of long-context LLMs that support effective context windows of up to 32,768 tokens. Our model series are built through continual pretraining from LLAMA 2 with longer training sequences and on a dataset where long texts are upsampled. We perform extensive evaluation on language modeling, synthetic context probing tasks, and a wide range of research benchmarks. On research benchmarks, our models achieve consistent improvements on most regular tasks and significant improvements on long-context tasks over LLAMA 2. Notably, with a cost-effective instruction tuning procedure that does not require human-annotated long instruction data, the 70B variant can already surpass `gpt-3.5-turbo-16k`'s overall performance on a suite of long-context tasks. Alongside these results, we provide an in-depth analysis on the individual components of our method. We delve into LLAMA's position encodings and discuss its limitation in modeling long dependencies. We also examine the impact of various design choices in the pretraining process, including the data mix and the training curriculum of sequence lengths – our ablation experiments suggest that having abundant long texts in the pretrain dataset is *not* the key to achieving strong performance, and we empirically verify that long context continual pretraining is more efficient and similarly effective compared to pretraining from scratch with long sequences.

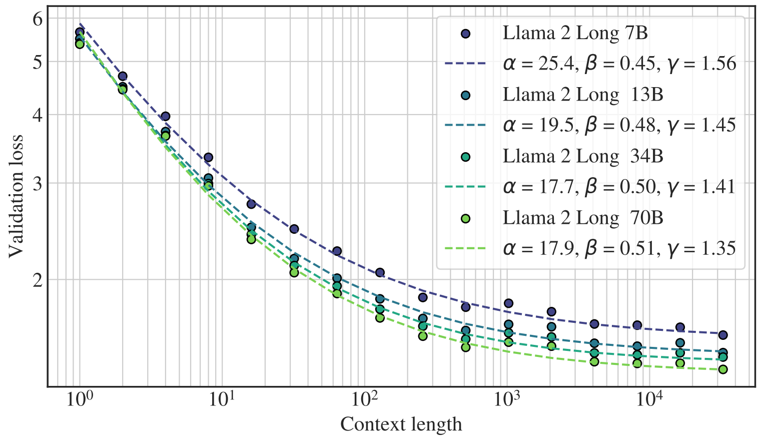


Figure 1: We show that our model's validation loss can be fit as a function of the context length: $L(c) = (\frac{\alpha}{c})^\beta + \gamma$ with a different set of α, β, γ for each model size. This power-law relationship also suggests that context length is another important axis of scaling LLMs and our model can continually improve its performance as we increase the context length up to 32,768 tokens.

1 Introduction

Large language models (LLMs), trained with an unprecedented magnitude of data and compute, hold the promise of fundamentally improving the way we interact with the digital world. As LLMs get rapidly deployed and continue to evolve through scaling, we envision these models to serve more intricate and complex use cases, such as analyzing dense knowledge-rich documents, powering more genuine and engaging chatbot experiences, and aiding human users in iterative creation processes such as coding and design, etc. A crucial feature supporting this evolution is the ability to effectively process long-context inputs.

Until now, LLMs with robust long-context capabilities are primarily provided through proprietary LLM APIs (Anthropic, 2023; OpenAI, 2023) and there is no open recipe for building long-context model that can demonstrate on-par downstream performance as these proprietary models. Moreover, existing open-sourced long-context models (Tworkowski et al., 2023b; Chen et al., 2023; Mohtashami and Jaggi, 2023; MosaicML, 2023b) often fall short on evaluations and primarily measure long-context capabilities with the language modeling loss and synthetic tasks, which do not comprehensively demonstrate their effectiveness in diverse, real-world scenarios. Additionally, these models often overlook the necessity of maintaining strong performance on standard short-context tasks, either bypassing the evaluations or reporting degenerated performance (Peng et al., 2023; Chen et al., 2023).

In this work, we describe our approach to build long-context LLMs with superior performance over all existing open-sourced models. We build our models by continually pretraining from LLAMA 2 checkpoints with additional 400 billion tokens formed as long training sequences. Among the model series, the smaller 7B/13B variants are trained with 32,768-token sequences while the 34B/70B variants with 16,384-token sequences. In contrast to the limited evaluation performed by existing studies, we extensively evaluate our models using language modeling, synthetic tasks, and also a wide range of real-world benchmarks covering both long and short context tasks. On language modeling, our model demonstrates a clear power-law scaling behavior with respect to context lengths. This scaling behavior, as shown in Figure 1, not only shows our models' ability to consistently benefit from more contexts but also suggest that context length is another importance axis of scaling LLMs. When comparing our models to LLAMA 2 on research benchmarks, we not only observe significant improvements on long-context tasks but also modest improvements on standard short-context tasks, especially on coding, math, and knowledge benchmarks. We explored using a simple and cost-effective procedure to instruction finetune our continually pretrained long models without any human-annotated data. The end result is a chat model that can achieve stronger overall performance than `gpt-3.5-turbo-16k` on a series of long-context benchmarks covering question answering, summarization, and multi-document aggregation tasks.

In the remaining part of this paper, we begin by presenting the continual long-context pretraining approach and a lightweight instruction tuning procedure, followed by detailed results on a range of short and long context tasks. To facilitate future research, we complement our results with an analysis section discussing how the design of positional encodings, the length distribution of the dataset and the training curriculum contributes to the final performance. Finally, we report responsible safety evaluations, which validates that our models can largely maintain the safety performance of the original LLAMA 2 series.

2 Method

2.1 Continual Pretraining

Training with longer sequence lengths can introduce significant computational overhead due to the quadratic attention calculations. This is the main motivation of our continual pretraining approach. The underlying hypothesis that similar long-context capabilities can be learned by continually pretraining from a short-context model is later validated in Section 4.4 through comparing different training curricula. We keep the original LLAMA 2 architecture nearly intact for continual pretraining and only make a necessary modification to the positional encoding that is crucial for the model to

[†] Equal contribution

[*] Corresponding authors:{xwhan, sinongwang, haom}@meta.com

attend longer. We also choose not to apply sparse attention (Child et al., 2019) in this work, since given LLAMA 2 70B's model dimension (h = 8192), the cost of attention matrix calculation and value aggregation only becomes a computation bottleneck when the sequence length exceeds 49,152 ($6h$) tokens (Narayanan et al., 2021).[1]

Positional Encoding Through early experiments at the 7B scale, we identified a key limitation of LLAMA 2's positional encoding (PE) that prevents the attention module from aggregating information of distant tokens. We adopt a minimal yet necessary modification on the RoPE positional encoding (Su et al., 2022) for long-context modeling – decreasing the rotation angle (controlled by the hyperparameter "base frequency b"), which reduces the decaying effect of RoPE for distant tokens. In Section 4.1, we show this simple method outperforms a concurrent approach (Chen et al., 2023) for extending LLAMA's context length and provide a theoretic explanation of its superiority.

Data Mix On top of the working model with the modified PE, we further explored different pretrain data mixes in Section 4.2 for improving long-context abilities, either by adjusting the ratio of LLAMA 2's pretraining data or adding new long text data. We found that often the quality of the data plays a more critical role than the length of texts for long-context continual pretraining.

Optimization Details We continually pretrain LLAMA 2 checkpoints with increased sequence length while keeping the same number of tokens per batch as in LLAMA 2. We train all models for a total of 400B tokens over 100,000 steps. With FLASHATTENTION (Dao et al., 2022), there is negligible GPU memory overhead as we increase the sequence length and we observe around 17% speed loss when increasing the sequence length from 4,096 to 16,384 for the 70B model. For the 7B/13B models, we use learning rate $2e^{-5}$ and a cosine learning rate schedule with 2000 warm-up steps. For the larger 34B/70B models, we find it important to set a smaller learning rate ($1e^{-5}$) to get monotonically decreasing validation losses.

2.2 Instruction Tuning

Collecting human demonstration and preference labels for LLM alignment is a cumbersome and expensive process (Ouyang et al., 2022; Touvron et al., 2023). The challenge and cost are more pronounced under long-context scenarios, which often involve complex information flow and specialized knowledge, e.g., processing dense legal/scientific documents, making the annotation task nontrivial even for skilled annotators. In fact, most existing open-source instruction datasets (Conover et al., 2023; Köpf et al., 2023) predominantly consist of short samples.

In this work, we found that a simple and cheap approach which leverages a pre-built large and diverse short-prompt dataset works surprisingly well on long-context benchmarks. Specifically, we take the RLHF dataset used in LLAMA 2 CHAT and augment it with synthetic self-instruct (Wang et al., 2022) long data generated by LLAMA 2 CHAT itself, in the hope that the model can learn a diverse set of skills through the large amount of RLHF data and transfer that knowledge to long-context scenarios via self-instruct data. The data generation process focuses on QA-format tasks: starting from a long document in our pretraining corpus, we select a random chunk and prompt LLAMA 2 CHAT to write question-answer pairs based on information in the text chunk. We collect both long and short form answers with different prompts. After that, we also adopt a self-critique step where we prompt LLAMA 2 CHAT to verify the model-generated answers. Given a generated QA pair, we use the original long document (truncated to fit the model's maximum context length) as the context to construct a training instance.

For short instruction data, we concatenate them as 16,384-token sequences. For long instruction data, we add padding tokens on the right so that models can process each long instance individually without truncation. While standard instruction tuning only calculates loss on the output tokens, we find it particularly beneficial to also calculate the language modeling loss on the long input prompts, which gives consistent improvements on downstream tasks (Section 4.3).

[1] While sparse attention might be useful for reducing the key/value cache size at inference time when trading off performance, it can complicate the inference pipeline and the improvements can also be offset by quantization methods.

Model	Size	Coding	Math	MMLU	Commonsense	OpenQA
Llama 2	7B	16.8	8.55	45.3	63.9	48.9
	13B	24.5	16.3	54.8	66.9	55.4
	34B	27.8	24.2	62.6	69.9	58.7
	70B	37.4	35.2	68.9	71.9	63.6
Llama 2 Long	7B	20.6	10.5	47.8	64.9	51.0
	13B	25.7	21.5	60.1	67.8	56.8
	34B	29.9	29.0	65.0	70.9	60.3
	70B	**39.9**	**41.3**	**71.7**	**72.7**	**64.0**

Table 1: Performance on standard short-context benchmarks. We report *Coding* score as the average of pass@1 of HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021); Math score as the average of top-1 accuracy of 8-shot GSM8K (Cobbe et al., 2021) and 4-shot MATH (Hendrycks et al., 2021); OpenQA score as the average of 5-shot performance on NaturalQuestions (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017); *Commonsense* score as the average of PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC easy and challenge (Clark et al., 2018), OpenBookQA (Mihaylov et al., 2018) and CommonsenseQA (Talmor et al., 2018).

Task	GPT-3.5	GPT-4	PaLM	PaLM-2-L	Llama 2	Llama 2 Long
MMLU (5-shot)	70.0	**86.4**	69.3	78.3	68.9	71.7
Natural Questions (1-shot)	-	-	29.3	**37.5**	33.0	35.7
GSM8K (8-shot)	57.1	**92.0**	56.5	80.7	56.8	65.4
HumanEval (0-shot)	48.1	**67.0**	26.2	-	29.9	32.9

Table 2: Comparison with closed models on standard short tasks.

3 Main Results

3.1 Pretrained Model Evaluation

Short Tasks To make long-context LLMs universally useful, an important desiderata is to ensure robust performance on standard short-context tasks. We verify our models' performance on a series of common benchmarks following the previous work (Touvron et al., 2023). The aggregated results are shown in Table 1. Overall, we observe *on-par and, in most cases, stronger results* than Llama 2. Notably, we observe significantly improved results on coding, math, and knowledge intensive tasks such as MMLU. As shown in Table 2, our model outperforms GPT-3.5 on MMLU and GSM8k. This is in contrast to a previous work (Chen et al., 2023) which observes degradation on short tasks. We attribute the improvements to additional computation FLOPs and the knowledge learned from newly introduced long data.

Long Tasks Different from previous works (Chen et al., 2023; Mohtashami and Jaggi, 2023) that mostly rely on perplexity and synthetic tasks to gauge long-context performance, we perform long-context evaluation using real-world language tasks. We evaluate 0-shot performance on NarrativeQA (Kočiský et al., 2018), 2-shot on QuALITY (Pang et al., 2022) and Qasper (Dasigi et al., 2021), and 1-shot on QMSum (Zhong et al., 2021). The number of shots are decided based on the average sample length of each dataset (i.e., samples in Qasper and QuALITY are often much shorter than those of NarrativeQA). We focus these QA-style tasks because of the ease of prompt engineering[2] and less biased automatic evaluations. The input prompts are truncated from the left side if the prompts exceed the maximum input length of the model or 16,384 tokens. We compare with open-source long-context models available in Huggingface Transformers, namely Focused Transformer (Tworkowski et al., 2023a), YaRN (Peng et al., 2023), Xgen (Nijkamp et al., 2023), MPT (MosaicML, 2023b,a) and Together's Llama 2 fork (Together, 2023). As shown in Table 3, our models achieve superior performance compared to these models. At the 7B scale, only "Together-7B-

[2] We use simple prompt "{CONTEXT} Q: {QUESTION}, A:" to evaluate all pretrained models.

Model	NarrativeQA F1 (0-shot)	Qasper F1 (2-shot)	QuALITY EM (2-shot)	QMSum ROUGE-geo* (1-shot)
Focused Transformer (3B)	16.3	15.4	20.5	10.6
Yarn-7B-128k	20.9	26.2	32.3	11.4
Together-7B-32k†	23.3	27.3	41.2	12.6
Xgen-7B-8k-base	17.4	20.5	21.0	6.79
MPT-7B-8k	18.8	24.7	23.7	8.78
Yarn-13B-128k	23.4	27.1	46.4	11.9
MPT-30B-8k	22.9	29.0	41.5	10.3
Llama 2 70B	25.7	27.5	53.0	11.9
Llama 2 Long 7B	21.9	27.8	43.2	14.9
Llama 2 Long 13B	25.6	31.2	57.6	15.7
Llama 2 Long 34B	29.4	33.7	65.7	15.9
Llama 2 Long 70B	**30.9**	**35.7**	**79.7**	**16.5**

Table 3: Comparison with open-source long-context models on research benchmarks. †: “together-7B-32k" is not a purely pretrained model and has been trained using supervised datasets which can improve its few-shot results. *: ROUGE-geo is the geometric mean of ROUGE-1, 2 and L. All numbers are validation results and the maximum allowed prompt length is set to 16,384 tokens.

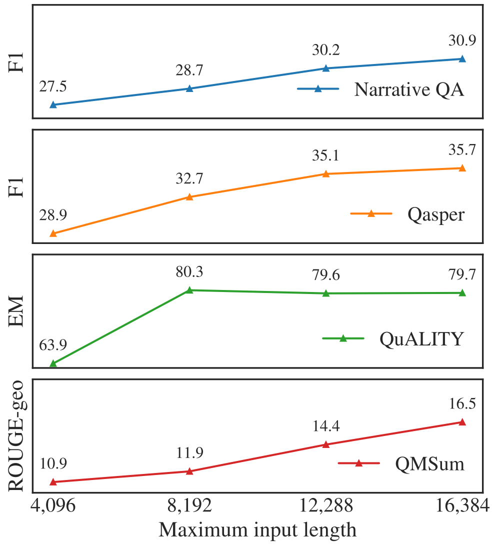


Figure 2: Performance on long-context tasks as the maximum context lengths of prompts increase.

32k” can match our model’s performance. Note that this model is not a purely self-supervised model and has been finetuned using a large supervised dataset to improve its few-shot results. As the 7/13B variants of our models have been trained with 32k-token sequences, we also perform comparisons using 32,768 maximum prompts lengths and the results are consistent, as shown in Table 13.

Effective Context Utilization To validate that our models can effectively use increased context window, we first show in Figure 2 that the results on each long task improve monotonically as we increase the context lengths. Inspired by (Kaplan et al., 2020; Hoffmann et al., 2022), we also found that the language modeling loss of our model follows a power-law plus constant scaling relationship with the context length (Figure 1), suggesting:

- Our model continues to show gain in performance (on the language modeling loss) up to 32,768 tokens of text, despite having diminishing returns. Taking our 70B model for example, if we double the context length, we can expect the loss to be reduced by a factor of $2^{-\beta} \approx 0.7$ plus a model specific constant $(1 - 2^{-\beta}) \cdot \gamma$.

- Larger models can leverage the contexts more effectively, indicated by the larger β value of the curves.

3.2 Instruction Tuning Results

We test our instruction tuned model on ZeroSCROLLS (Shaham et al., 2023) which bundles 10 long-context datasets spanning from summarization, question answering, to multi-document aggregation tasks. For a fair comparison, we use the same configuration (prompts, truncation strategy, and maximum generation lengths, etc) as specified by the benchmark. As shown in Table 4, without using any human annotated long context data, our 70B chat model is able to outperform `gpt-3.5-turbo-16k` on 7 out of the 10 tasks. In addition, we run evaluations on six new long tasks introduced in L-Eval (An et al., 2023) and again observe strong results, as shown in Table 17 in the Appendix. We see that the finetuned model is particularly good at QA tasks which is the main theme of the self-instruct data. We expect the performance to be further improved if more diverse data are used for finetuning.

It is worth mentioning that evaluating long-context LLMs is a nontrivial task. The automatic metrics used in these benchmarks are limited in many ways. For instance, the summarization tasks only come with a single ground-truth summary and the n-gram matching metrics do not necessarily align with human preference. For QA and aggregation tasks, where the metric is less of a concern, truncating the input context might also remove the information necessary to answer the question. Another important caveat is that most proprietary models do not share their training data details, which makes it hard to take into consideration the potential leakage during public benchmark evaluation.

Model	Summarization			Question answering					Aggregation		
	GR	SS	QM	SQAL	Qspr	Nrtv	QALT	MuSQ	SpDg	BkSS	Avg
GPT-3.5-turbo (4k)	21.3	16.1	15.6	20.4	49.3	25.1	66.6	27.1	49.1	49.8	34.0
GPT-3.5-turbo-16k†	24.3	16.2	17.4	21.4	50.0	29.5	72.0	27.0	54.1	54.6	36.7
Claude (8k)	24.2	16.1	14.6	21.0	52.3	32.6	84.8	36.1	61.6	47.4	39.1
GPT4 (8k)	26.3	17.3	18.5	22.6	50.7	27.6	89.2	41.1	62.8	60.5	41.7
LLAMA 2 LONG CHAT 70B	26.0	15.0	20.0	20.9	52.0	31.7	82.6	27.3	55.5	46.2	37.7

Table 4: ZeroSCROLLS long-context leaderboard results. †Evaluated as of 8/7/2023. The GPT-4 and Claude results are directly copied from the leaderboard. Underscored are the 7/10 tasks where our model outperforms `gpt-3.5-turbo-16k`.

3.3 Human Evaluation

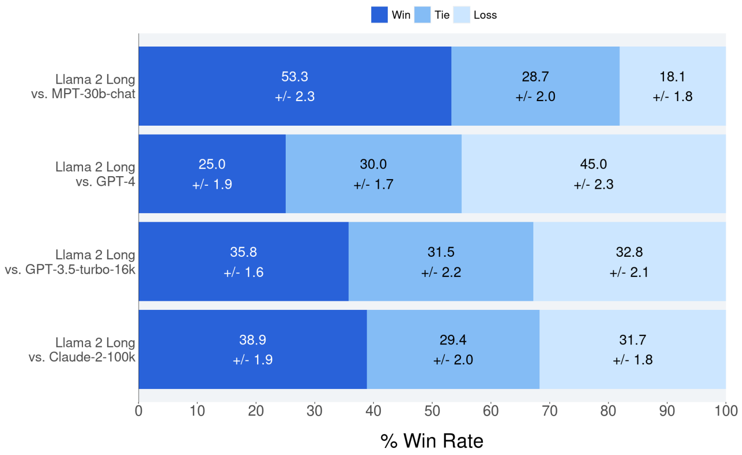


Figure 3: Human preference on model responses with multi-turn conversation and multi-document search query answering data.

Complementary to the automatic evaluation benchmark results, we conduct human evaluations by asking annotators whether they prefer the generation from our instruction finetuned model or from proprietary models like MPT-30B-chat, GPT-4, GPT-3.5-turbo-16k, and Claude-2 in terms of helpfulness, honesty, and harmlessness. Unlike automatic metrics, humans are better at evaluating the quality of model responses for long context models because of the large space of acceptable answers. We focus on two major application scenarios with a total of 2,352 examples. For multi-turn conversation data, each prompt is a chat history based on which the model needs to generate a coherent response. For the multi-document search query answering application, the model is provided with a few most relevant documents retrieved from a search session and the corresponding search query. We then evaluate how well these models can leverage the information (retrieved documents) to answer the given query. Each comparison example was evaluated by 3 different human annotators. The standard win rate of our our model over each model is calculated by averaging the result of each comparison example and the final score along with the 95% confidence interval is shown in Figure 3. With very little instruction data, our model can achieve competitive performance against MPT-30B-chat, GPT-3.5-turbo-16k, and Claude-2. It is worth noting that human evaluation on longer context tasks is challenging and generally requires well trained and skilled annotators. We hope this study can not only give a sense of the potential of our instruction finetuned model on some long context downstream applications but also motivate future efforts in developing more robust long context automatic evaluations.

4 Analysis

In this section. We perform ablation experiments to justify our design choices (i.e. architecture modification, data mixes, and training curriculum) and quantify their contributions to the final performance.

4.1 Positional Encoding for Long Text

Our early experiments used a synthetic "FIRST-SENTENCE-RETRIEVAL" task to probe the effective context window of the pretrained models where we simply prompt the model to return the first sentence of the input. Our initial task results suggest that, with the original LLAMA 2 architecture untouched, our model was unable to effectively attend beyond $4,000$ - $6,000$ tokens even after extensive long-context continual pretraining. We hypothesize that this bottleneck comes from the ROPE positional encoding used in LLAMA 2 series which imposes a heavy decay on the attention scores[3] for distant tokens. We propose a simple modification to the default RoPE encoding to reduce the decaying effect – increasing the "base frequency b" of ROPE from $10,000$ to $500,000$, which essentially reduces the rotation angles of each dimension. The idea is also independently and concurrently suggested in the Reddit r/LocalLLaMa community. The effect of the base frequency change is visualized in Figure 4. Another concurrent approach named "position interpolation" (PI) (Chen et al., 2023) proposes to linearly scale the input positions such that the positions of tokens in the long sequences will to mapped to the model's original position range. As shown by the figure, it also implicitly achieves a decay reduction effect.

PE	Books	CC	Wikipedia
RoPE	6.548	6.816	3.802
ROPE PI	6.341	6.786	3.775
ROPE ABF	**6.323**	**6.780**	**3.771**
XPOS ABF	6.331	**6.780**	**3.771**

Table 5: Validation perplexity of models with different positional encoding variants. All samples are $32,768$-token sequences (CC: CommonCrawl).

Another interesting observation from the visualization is that RoPE introduces large "oscillation" in the long-range regions, which could be undesirable for language modeling (Sun et al., 2022). To investigate whether this effect hurts performance, we also explored another recently proposed variant of rotary encoding, XPOS (Sun et al., 2022), which smooths the high-frequency component. Note that XPOS with the default parameters suffers from the same decaying issue as ROPE and therefore, we also applied a similar decay fix to XPOS.

Specifically, we empirically compare the following methods: the ROPE baseline, PI, our proposed RoPE with adjusted base frequency (denoted as ROPE ABF), and XPOS ABF (visual comparisons in Figure 4). We report results on 1) long-sequence validation perplexity in Table 5 and Figure 5a, 2) the

[3] The quantity that heavily decays is $\mathbb{E}_{q,k}[\text{RoPE}(q,m)^\top \text{RoPE}(k,n)|m,n]$ as the relative position $|m-n|$ gets larger where q, k are the query and key of the two tokens at position m and n.

	HumanEval	Math	MMLU	HellaSwag	TQA
RoPE	14.63	**3.62**	45.69	76.31	65.23
RoPE PI	15.24	3.08	45.84	76.65	65.96
RoPE-ABF	**17.07**	3.52	**46.24**	**76.73**	66.04
xPos-ABF	16.46	3.54	45.72	76.68	**66.14**

Table 6: The performance of models with different positional encoding variants on standard short-context benchmarks.

FIRST-SENTENCE-RETRIEVAL context probing task[4] in Figure 5b, and 3) some representative regular context tasks in Table 6 (to validate that long models do not degenerate on short-context tasks). All model variants are continually pretrained from the 7B LLAMA 2 checkpoint with additional 80B tokens organized as 32,768-token long sequences.

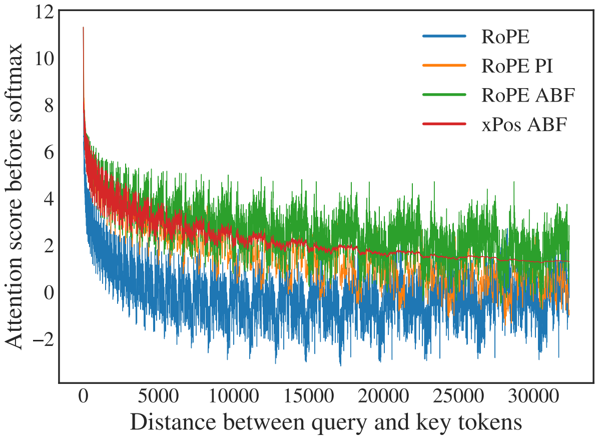


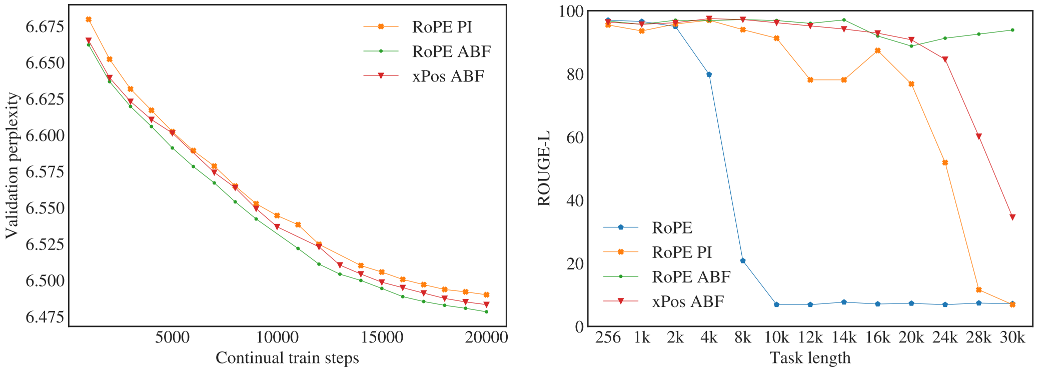
Figure 4: Decaying raw attention scores for distant tokens of explored positional encoding variants (assuming keys and queries are all-ones vectors).

Overall, results on these evaluations suggest that ROPE ABF performs the best among all explored variants. In particular, we see that ROPE ABF is the only variant that can maintain its performance up to the full 32,768-token context window on the FIRST-SENTENCE-RETRIEVAL task. We also found that XPOS ABF with less oscillation does not lead to substantial gains, suggesting that these artifacts are not detrimental to language modeling. While XPOS is claimed to possess better extrapolation property (Sun et al., 2022), we found that, with the base frequency modification, XPOS does not extrapolate better than ROPE (see Appendix C). In addition to empirical results, we provide a theoretical analysis of RoPE ABF and its difference to PI in Appendix B. We argue that RoPE ABF distributes the embedded vectors with an increased granularity when compared to RoPE PI, making it a easier for the model to distinguish between positions. It is worth noting that the relative distance between the embedded vectors has a linear dependence on the key parameter of RoPE PI and a logarithmic dependence on the key parameter of RoPE ABF, which coincides with our empirical observation that the base-frequency is not very sensitive and can be easily adjusted based on the max sequence length.

4.2 Pretraining Data Mix

The data used to continually pretrain our model combines existing datasets used by LLAMA 2 and new long text data. We also adjusted the data source mix ratio to up-weight long data samples. Our early experiments with 7B models confirms the significant improvements using this data mix for

[4] We also test on the PASSKEY task as used in (Mohtashami and Jaggi, 2023). All the model variants except ROPE can achieve perfect accuracy. We believe this task is overly simple for context probing.



(a) Validation PPL (16k-token sequences) on a held-out long-context dataset.

(b) Performance on FIRST-SENTENCE-RETRIEVAL task.

Figure 5: Comparison of positional encoding variants on synthetic sentence retrieval task and validation perplexity evolution during continual pretraining.

Continual Pretrain Data	NarrativeQA Δ F1	Qasper Δ F1	Quality Δ EM	QMSum Δ ROUGE-geo
LLAMA 2 LONG data mix	**23.70%**	**43.64%**	**75.5%**	**45.70%**
LLAMA 2 data mix	18.23%	38.12%	60.3%	44.87%
- remove long text data	19.48%	39.14%	67.1%	36.60%
- upsample existing long text data	22.15%	36.82%	65.0%	42.83%

Table 7: Comparison of different pretraining data mix on long-context tasks. Instead of showing the absolute performance, we report relative improvements over the 7B LLAMA 2 which has a 4,096-token context window. All models are evaluated with prompts truncated at 16,384 tokens.

long-context tasks, as shown in the first two rows of Table 7. In this section, we aim to rigorously investigate the source of improvements. In particular, we are interested in differentiating the effects of the data length distribution and the quality of the corpus itself.

We perform two additional ablations using LLAMA 2's pretrain datasets: 1) we remove the long text data from the LLAMA 2 dataset and continually pretrain our model with mostly short documents; 2) we increase the sample weights of existing long text data to be similar to the long text ratio used by proposed new model. Interestingly, even with most of the long texts removed, the model can still obtain most of the performance gain over LLAMA 2. We also find that there is no clear and consistent advantage as we greatly increase the long data ratio (the third row v.s. the fourth row in Table 7 and Table 8). We observe similar results on the FIRST-SENTENCE-RETRIEVAL task as shown by Figure 7 in the Appendix.

Based on the above ablations, we can see that adjusting the length distribution of the pretrain data does not provide major benefits. However, as we evaluate these model variants' performance on standard short-context tasks, we find that new data mix also leads to large improvements in many cases, especially knowledge-intensive tasks like MMLU, as shown in Table 8. These results suggest that *long-context LLMs can be effectively trained even with very limited long data* and the improvements of our pretrain data over the one used by LLAMA 2 mostly come from the quality of the data itself, instead of the length distribution difference.

Continual Pretrain Data	HumanEval	Math	MMLU	HellaSwag	TQA
LLAMA 2 LONG data mix	**17.08**	**4.09**	**48.62**	76.74	66.24
LLAMA 2 data mix	15.24	3.61	46.30	76.63	**66.71**
- remove long text data	17.07	3.57	46.25	**76.76**	65.90
- upsample existing long text data	17.07	3.53	46.25	76.74	66.04

Table 8: Standard short task performance of long-context models with different pretrain data mix.

4.3 Instruction Tuning

We explored various strategies to instruction-finetune the pre-trained long context model which do not require any supervised long data. We start with only finetuning the models with short instruction data from LLAMA 2 CHAT (referred as "RLHF V5" in (Touvron et al., 2023)) and then blend in with some pretrain data to avoid forgetting of previous long context continual pretraining. As demonstrated in Table 9, using only short instruction data can already produce a decent long model that significantly outperforms LLAMA 2 on various long-context tasks. On top of this dataset that only includes short prompts, we see that adding pretrain data (calculating language modeling loss on the whole sequence) can further boost the performance on most datasets. Inspired by this, we add the LM loss over the long context inputs when we finetune with self-instruct data. This simple trick makes learning more stable when we have unbalanced input and output lengths[5], which gives significant improvements on most of the tested tasks (the last two rows of Table 9).

Settings	Qasper	NarrativeQA	QuALITY	SummScreenFD	QMSum
LLAMA 2 CHAT baseline	12.2	9.13	56.7	10.5	14.4
LLAMA 2 LONG *finetuned* with:					
"RLHF V5"	22.3	13.2	71.4	14.8	16.9
"RLHF V5" mix pretrain	23.7	16.6	76.2	**15.7**	17.8
"RLHF V5" mix self-inst w/o LM loss	35.7	22.3	59.3	12.2	13.4
"RLHF V5" mix self-inst with LM loss	**38.9**	**23.3**	**77.3**	14.5	**18.5**

Table 9: Comparison of different instruction finetuning data mixes.

4.4 Training Curriculum

Continual pretraining has demonstrated its efficacy in our experiments, but an open question still remains: does pretraining from scratch with long sequences yield better performance than continual pretraining? In this section, we study different training curricula and try to investigate if continual pretraining can offer competitive performance with less computation budget. We start off by pretraining a 7B model with 32,768 sequence length from start to the end. Then we explored various two-stage training curricula where we begin with 4096 sequence length and switch to 32,768 when the model completes 20%, 40%, 80% of whole training process. For all cases, we keep the same number of total training tokens and make sure the number of tokens per each gradient update remains constant (4 million tokens) by adjusting the batch size and sequence length accordingly.

We evaluate our models on the long-text QA tasks used in Section 4.2 and report the final models' perplexity on different validation corpora. As shown in Table 10 and Table 11, continual pretraining from short context models can easily save around 40% FLOPs while imposing almost no loss on performance. These results also align with the training loss curves we observed from each run in Figure 6 – the models can quickly adapt to the increased sequence length and get to similar loss scale.

Pretrain Curriculum	**FLOPs**	NarrativeQA F1	Qasper F1	Quality EM	QMSum ROUGE-geo
32k from scratch	3.783×10^{22}	18.5	28.6	37.9	11.46
4k→32k @ 20%	3.405×10^{22}	20.0	28.1	38.8	12.09
4k→32k @ 40%	3.026×10^{22}	20.1	27.0	37.4	12.44
4k→32k @ 80%	2.270×10^{22}	18.5	25.0	38.3	11.00

Table 10: Comparison of models with different training curricula on long context QA tasks.

[5] In our cases, the output lengths of most samples are a lot shorter than the those of the long-context inputs.

Model	CC	Books	Wikipedia
32k from scratch	7.67	6.52	4.31
4k→32k @ 20%	7.59	6.46	4.26
4k→32k @ 40%	7.59	6.46	4.25
4k→32k @ 80%	7.59	6.49	4.25

Table 11: Perplexity evaluation of models with different training curricula on three validation sets.

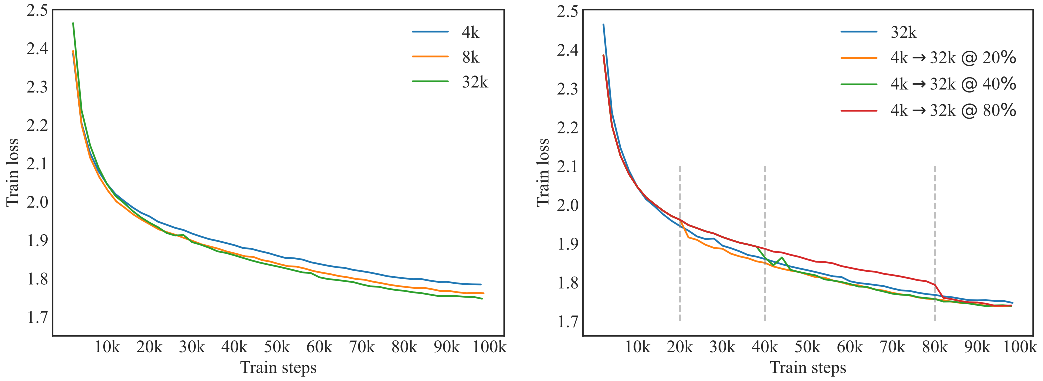


Figure 6: Smoothed loss curves for the training curriculum ablation. On the left, we show losses for models trained with a fixed context window. On the right, we compare training curricula where we switch the context length from 4,096 to 32,768 tokens at different stages indicated by the dashed lines. Our models can quickly adapt to the new sequence length within a few thousand steps.

5 AI Safety

5.1 Evaluation on Safety Benchmarks

Despite showing excellent performance on various of downstream tasks, large language models are prone to generating harmful, misinformative, and biased contents (Lin et al., 2021; Hartvigsen et al., 2022; Dhamala et al., 2021; Ji et al., 2023). Long-context language models can process extended inputs in their context window, but at the same time, they also face a higher risk of jailbreak, especially through means such as prompt injection (Greshake et al., 2023). In this section, we evaluate the safety capability of instruction fine-tuned model using three standard academic benchmarks including TruthfulQA (Lin et al., 2021), ToxiGen (Hartvigsen et al., 2022), and BOLD (Dhamala et al., 2021), similar to (Touvron et al., 2023). We focus on the largest instruction fine-tuned model variant (i.e., 70B) and compare its results with both open sourced LLMs (Falcon-instruct Almazrouei et al. (2023), MPT-instruct MosaicML (2023a)) and propriety LLMS (GPT-3.5, GPT-4 (OpenAI, 2023), Claude-2 (Anthropic, 2023)) in Table 12.

We observe that in general instruction fine-tuned model maintains similar safety performance compared to LLAMA 2 CHAT and is safer and less biased compared to other open-source LLMs such as Falcon-instruct and MPT-instruct. AI safety is a complex domain and it can be extremely difficult to comprehensively evaluate all safety aspects of instruction fine-tuned model with three benchmarks. However, we hope our analysis can serve as a pilot study and provide directional signals on long-context large language models' safety performance, which are not discussed in other works on the same topic (Tworkowski et al., 2023b; Ding et al., 2023; Chen et al., 2023). Currently the community also lacks dedicated safety benchmarks for long-context large language model evaluation and we plan to invest in this direction in our future work.

TruthfulQA We evaluate instruction fine-tuned model on TruthfulQA (Lin et al., 2021) to benchmark its factuality. The benchmark consists of 817 questions covering 38 categories including health, law, finance, and politics (Lin et al., 2021). Similar to (Touvron et al., 2023), we use few-shot prompts with 6 random QA pairs for generation and then leverage two fine-tuned GPT-3 models to classify whether the generation is truthful and informative. We report the percentage of generations that are both truthful and informative as the final metric in Table 12.

ToxiGen We measure the toxicity of instruction fine-tuned model using ToxiGen (Hartvigsen et al., 2022) where we check the percentage of toxic and hateful generations against 13 minority groups. Following (Touvron et al., 2023), we filtered out prompts where annotators disagree with each other on the target demographic group. We use the default ToxiGen classifier fine-tuned based on RoBERTa (Liu et al., 2019) to evaluate the level of toxicity of the model's outputs. We report the percentage of toxic generations across all groups in Table 12.

BOLD Bias in Open-Ended Language Dataset (BOLD) Dhamala et al. (2021) is used in this work to quantify how biased the models are against people from different demographic groups. This dataset consists of 23,679 prompts extracted from English Wikipedia covering five domains including race, gender, religion, political ideology and profession with 43 subgroups in total. Following Touvron et al. (2023), we exclude prompts belonging to Hinduism and Atheism religious subgroups as they only feature 12 and 29 prompts, respectively. After generations are inferred from each model, we leverage the Valence Aware Dictionary and Sentiment Reasoner (VADER) Hutto and Gilbert (2014) to perform sentiment analysis with a score ranging between -1 and 1. A positive score corresponds to a positive sentiment towards the subgroup mentioned in the prompt and vice versa. A sentiment score close to 0 indicates neutral sentiment which is desired. We report the average sentiment score across 43 demographic subgroups as the final metric for BOLD in Table 12.

	Model Size	TruthfulQA ↑	ToxiGen ↓	BOLD ↓
GPT-3.5-turbo	-	78.46	0.01	0.50
GPT-3.5-turbo-16k	-	75.15	0.07	0.49
Claude-2	-	62.66	0.05	0.46
GPT4	-	**80.66**	0.03	0.43
Falcon-instruct	40B	57.41	3.3	0.39
MPT-instruct	30B	42.71	16.85	**0.34**
LLAMA 2 CHAT	70B	64.14	0.01	0.41
LLAMA 2 LONG CHAT	70B	60.95	**0.00**	0.40

Table 12: Evaluation of fine-tuned LLMs on three safety benchmarks. For TruthfulQA, we present the percentage of generations that are both truthful and informative (the higher the better). For ToxiGen, we present the percentage of toxic generations across all groups (the smaller the better). For BOLD, we report the average sentiment score across 43 demographic groups (the closer to 0 the better).

5.2 Red Teaming Exercises

Currently there is no open-sourced safety benchmark designed for long-context understanding. To ensure that the models are safe in long context use scenarios, we performed internal red teaming to better understand the vulnerability of our chat model. We attack the model by feeding long contexts (e.g., long conversations) to it, followed by adversarial prompts covering risky areas including illicit and criminal conducts (e.g., terrorism, theft, and human trafficking), hateful and harmful behaviors (e.g., defamation, self-harm, eating disorders, and discrimination), and unqualified advice Touvron et al. (2023). Through manual inspection, we did not observe significant risks compared to LLAMA 2 CHAT Touvron et al. (2023). We plan to invest more in new attack vectors against long context large models in future work.

6 Limitations

Limited Functionality. The our model proposed in this paper has not yet been finetuned for a wide range of long-context applications, such as creative writing that require long-form outputs. Applying existing alignment recipes, e.g., RLHF, for various scenarios is expensive and nontrivial. Even skilled annotators may struggle to the intricate details in dense texts. In this regard, we consider developing efficient alignment methods for long LLMs to be a very valuable direction for future research.

Tokenizer Efficiency. While the proposed our model series can consume contexts up to 32,768 tokens, the actually number of words our model can take is largely affected by the tokenizer behaviour. The tokenizer used by the Llama series has a relatively small vocabulary (32k symbols) and often produces longer sequences compare to the sequences given by GPT-3.5's tokenizer – we observe our tokenizer often produce 10% more tokens on average. Additionally, the tokenizer we use also cannot efficiently handle whitespace, making it inefficient to process long code data.

Hallucination. Like other LLMs, we have observed hallucination issue when testing the proposed our model. While this issue is common for short-context models, tackling with this problem for long-context models can be more pronounced because of the dense information they consume and the insufficient alignment process.

7 Conclusion

We present a series of long-context LLMs that leverage a simple yet necessary position encoding refinement and continual pretraining to achieve strong long-context performance. Our long context scaling is performed by continually pretraining from LLAMA 2 with additional 400B tokens and outperform LLAMA 2 on both short and long-context tasks. Our models also demonstrate superior performance compared to existing open-source long-context models and compare favorably against `gpt-3.5-turbo-16k` on a suite of long-context tasks after a simple instruction finetuning procedure without human supervision. We complement our results with a comprehensive analysis, providing insights on the influences of various factors including the nuances of position encodings, the data mix, and the pretraining curriculum on the final performance. We hope our study could make long-context LLMs more accessible and facilitate further advancements in this field.

8 Acknowledgement

We would like to thank Nikolay Bashlykov, Matt Wilde, Wenyin Fu, Jiangyu Huang, Jenya Lee, Mathew Oldham, and Shawn Xu for their invaluable support on the data, infrastructure, and various other aspects of this project.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. Falcon-40B: an open large language model with state-of-the-art performance. 2023.

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

Anthropic. Introducing 100K Context Windows, 2023. URL `https://www.anthropic.com/index/100k-context-windows`.

Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. Program synthesis with large language models. *arXiv:abs/2108.07732*, 2021.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439, 2020.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation, 2023.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

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

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023. URL `https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm`.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *NeurIPS*, 2022.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.365. URL `https://aclanthology.org/2021.naacl-main.365`.

Jwala Dhamala, Tony Sun, Varun Kumar, Satyapriya Krishna, Yada Pruksachatkun, Kai-Wei Chang, and Rahul Gupta. Bold: Dataset and metrics for measuring biases in open-ended language generation. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 862–872, 2021.

Jiayu Ding, Shuming Ma, Li Dong, Xingxing Zhang, Shaohan Huang, Wenhui Wang, and Furu Wei. Longnet: Scaling transformers to 1,000,000,000 tokens, 2023.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. *arXiv preprint arXiv:2302.12173*, 2023.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. *arXiv preprint arXiv:2203.09509*, 2022.

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

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022.

Clayton Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 216–225, 2014.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12): 1–38, 2023.

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

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018. doi: 10.1162/tacl_a_00023. URL `https://aclanthology.org/Q18-1023`.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. Openassistant conversations– democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*, 2023.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

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

MosaicML. Introducing mpt-30b: Raising the bar for open-source foundation models, 2023a. URL `www.mosaicml.com/blog/mpt-30b`. Accessed: 2023-06-22.

MosaicML. Introducing mpt-7b: A new standard for open-source, ly usable llms, 2023b. URL `www.mosaicml.com/blog/mpt-7b`.

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, et al. Efficient large-scale language model training on gpu clusters using megatron-lm. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15, 2021.

Erik Nijkamp, Tian Xie, Hiroaki Hayashi, Bo Pang, Congying Xia, Chen Xing, Jesse Vig, Semih Yavuz, Philippe Laban, Ben Krause, et al. Long sequence modeling with xgen: A 7b llm trained on 8k input sequence length. *Salesforce AI Research Blog*, 2023.

OpenAI. Gpt-4 technical report, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. QuALITY: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5336–5358, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.391. URL `https://aclanthology.org/2022.naacl-main.391`.

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

r/LocalLLaMa. NTK-Aware Scaled RoPE allows llama models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation. `https://www.reddit.com/r/LocalLLaMA/comments/14lz7j5/ntkaware_scaled_rope_allows_llama_models_to_have/`. Accessed: 2023-08-25.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. Zeroscrolls: A zero-shot benchmark for long text understanding, 2023.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2022.

Yutao Sun, Li Dong, Barun Patra, Shuming Ma, Shaohan Huang, Alon Benhaim, Vishrav Chaudhary, Xia Song, and Furu Wei. A length-extrapolatable transformer, 2022.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.

Together. Llama-2-7b-32k-instruct — and fine-tuning for llama-2 models with together api, 2023. URL https://together.ai/blog/llama-2-7b-32k-instruct.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Szymon Tworkowski, Konrad Staniszewski, Mikołaj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Miłoś. Focused transformer: Contrastive training for context scaling, 2023a.

Szymon Tworkowski, Konrad Staniszewski, Mikołaj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Miłoś. Focused transformer: Contrastive training for context scaling, 2023b.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence?, 2019.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.472. URL https://aclanthology.org/2021.naacl-main.472.

A More Results

Model	Prompt length	NarrativeQA F1 (0-shot)	Qasper F1 (2-shot)	QuALITY EM (2-shot)	QMSum ROUGE-geo* (1-shot)
Yarn-7B-128k	16k	20.9	26.2	32.3	11.4
Together-7B-32k	16k	23.3	27.3	41.2	12.6
Yarn-13B-128k	16k	23.4	27.1	46.4	11.9
Yarn-7B-128k	32k	24.0	26.2	30.4	13.6
Together-7B-32k	32k	24.7	27.3	41.3	14.2
Yarn-13B-128k	32k	25.5	27.1	48.0	13.8
Llama 2 Long 7B	16k	21.9	27.8	43.2	14.9
Llama 2 Long 13B	16k	25.6	31.2	57.6	15.7
Llama 2 Long 7B	32k	24.4	28.7	43.6	15.9
Llama 2 Long 13B	32k	**27.4**	**31.6**	**59.0**	**17.0**

Table 13: Comparison of our models with open-source long-context models on research benchmarks *using a maximum prompt length of 32,768 tokens*.

Model	Humanities	STEM	Social Sciences	Other
Llama 2 Long 7B	54.8	35.7	58.4	53.2
Llama 2 Long 13B	69.0	44.4	71.3	65.8
Llama 2 Long 34B	73.5	49.9	78.4	69.3
Llama 2 Long 70B	80.1	55.5	84.4	74.9

Table 14: Decomposed MMLU results.

Model	HumanEval	MBPP	MATH	GSM8k	NQ	TQA
Llama 2 Long 7B	18.3	23.0	4.22	16.8	27.5	74.4
Llama 2 Long 13B	19.5	31.8	8.38	34.6	32.5	81.1
Llama 2 Long 34B	22.6	37.2	10.6	47.4	35.0	85.6
Llama 2 Long 70B	32.9	46.8	17.2	65.4	39.8	88.2

Table 15: Results on HumanEval (0-shot), MBPP (3-shot), MATH (4-shot), GSM8K (8-shot), NaturalQuestions (5-shot) and TriviaQA-wiki (5-shot).

Model	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	CSQA
Llama 2 Long 7B	78.9	48.7	77.8	70.4	76.2	52.0	59.0	61.0
Llama 2 Long 13B	81.6	50.7	81.2	74.1	77.7	51.4	55.6	70.4
Llama 2 Long 34B	82.6	51.7	83.8	77.5	79.7	54.8	60.2	77.0
Llama 2 Long 70B	83.3	52.8	85.7	79.6	80.3	58.4	59.6	81.9

Table 16: Commonsense reasoning decomposed results. We use the same number of shots and evaluation metrics for all tasks as Llama 2.

B Theoretical Analysis of Positional Encodings

RoPE maps an argument vector $x \in \mathbb{R}^d$ into the embedding curve on a sphere in $\mathbb{C}^{d/2}$ parametrized by a real parameter $t \in \mathbb{R}$ and "base frequency" b:

$$f^{RoPE}(x,t)_j = (x_{2j} + i x_{2j+1})\, e^{i b^{-\frac{2j}{d}} t}.$$

Model	Coursera	TPO	TopicRetrieval	FinQA	ContractQA	NaturalQuestions
Claude 1.3 100k	60.2	83.6	70.6	-	-	-
gpt-3.5-turbo-16k	59.7	69.9	69.3	45.4	24.9	45.9
Best open models reported in An et al. (2023)						
longchat-13b-16k	36.8	55.4	33.3	37.9	21.1	22.8
chatglm2-6b-8k	47.2	54.6	10.0	34.8	16.4	17.6
LLAMA 2 LONG CHAT	52.9	81.8	76.0	47.3	25.5	66.7

Table 17: Evaluation on additional long-context tasks from L-Eval. We report the official metrics defined in An et al. (2023) and the results of compared models are directly token from the paper.

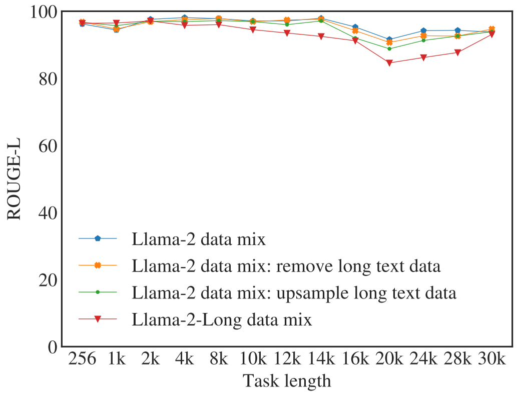


Figure 7: FIRST-SENTENCE-RETRIEVAL performance of models trained with different data mixes.

The purpose of this mapping is to help the attention module to separate the vectors corresponding to two instances of the same token that are situated at different positions in the input sequence.

Aiming at extending the sequence length of a transformer pretrained with a particular positional embedding f from L to $\hat{L}$, we would like to come up with a positional embedding $\hat{f}$ that minimizes the distance between the old and the new images of the embedded vectors:

$$d(f, \hat{f}) = \max_{x \in \mathcal{X}} \min_{k \in \{0,..N-1\}} \min_{j \in \{0,..\hat{N}-1\}} \text{dist}[f(x,k), \hat{f}(x,j)],$$

where $\mathcal{X} \subset \mathbb{R}^d$ is the set of vectors that would need to be positionally embedded. (Chen et al., 2023) computed this distance through the magnitude of the attention scores, but still argued for the efficiency of their method “position interpolation”) due to its reduced value of the distance to the original RoPE images when compared to the naive extrapolation of the positional embedding.

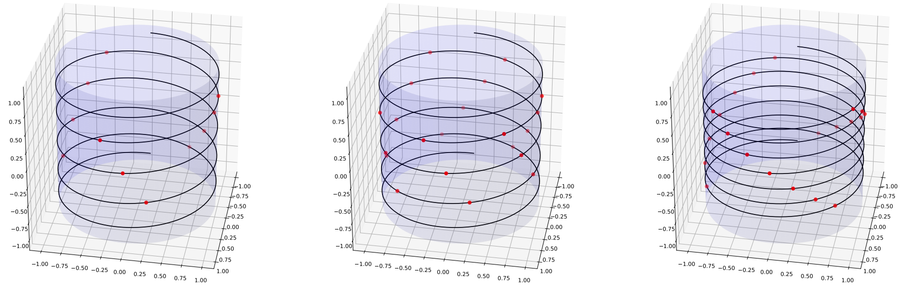
With this in mind, we consider two different methods to extend the sequence length of a trained transformer: Position Interpolation (PI) parameterized with α, and Adjusted Base Frequency (ABF) parameterized with β. These two methods correspond to the following embedding curves:

$$f^{RoPE+PI}(x,t)_j = (x_{2j} + i x_{2j+1})\, e^{i\alpha \cdot (b^{-\frac{2j}{d}})t}$$

$$f^{RoPE+ABF}(x,t)_j = (x_{2j} + i x_{2j+1})\, e^{i(\beta b)^{-\frac{2j}{d}}t}$$

Evaluating a positional embedding a-priori, we should consider the degree of granularity with which the embedding images are being distributed over the embedding space. Comparing alternative positional embeddings $\hat{f}$ mapping $\mathbb{R}^d \times \mathbb{N}$ into $\mathbb{C}^{d/2}$, we should prefer the one with the maximal value of the distance between the two closest images:

$$q(\hat{f}) = \min_{x \in \mathcal{X}; k \neq j \in \{0..\hat{N}-1\}} \text{dist}[\hat{f}(x,k), \hat{f}(x,j)].$$



(a) RoPE (b) RoPE+PI (c) RoPE+ABF

Figure 8: RoPE variants visualization as helices.

This leaves us with a multi-objective decision selecting the positional embedding for a model with extended context: on one hand, $\hat{f}$ should be chosen so that it minimizes $d(f, \hat{f})$, while on the other hand its value of $q(\hat{f})$ should be big enough.

Before proceeding to the explanation on how we make this multi-objective decision, we would like to provide a geometric intuition for the positional embeddings considered here. While it is difficult to visualize a mapping $\mathbb{R}^d \times \mathbb{N} \to \mathbb{C}^{d/2}$, we can consider $x \in \mathbb{R}^d$ to be fixed and visualize the projection $\mathbb{R} \to \mathbb{R}^3$. To get the intuition behind PI and ABF, let us consider the helix that is formed by $\mathrm{Re}\big[f^{RoPE}(x,t)_0\big]$, $\mathrm{Im}\big[f^{RoPE}(x,t)_0\big]$ and $\mathrm{Re}\big[f^{RoPE}(x,t)_j\big]$. The example on the Figure 8a depicts a black helix line given with the system

$$x = \cos t; y = \sin t; z = \sin at.$$

The red dots on the line correspond to 11 integer values of t.

Figure 8b aims to illustrate the impact of Position Interpolation on the relative position of the mapped vectors. The distance between the consecutive points got reduced considerably compered to Figure 8a. The impact of Adjusted Base Frequency is illustrated on Figure 8c. The distance between the consecutive points remained almost the same as on Figure 8a, although the minimal distance between points got considerably reduced due to the increased frequency of the helix. This effect of increased frequency of the helix would be reduced in the high dimension setting. The value of the coefficient a for the helix depicted on Figure 8a is two times larger than the value of the coefficient a for the helix depicted on Figure 8c. If the dimension of the input of the attention mechanism is $d = 128$, then the difference between $\theta_1 = b^{-\frac{2j}{d}}$ at $b = 10,000$ and $\theta_1 = b^{-\frac{2j}{d}}$ at $b = 500,000$ is only 6%. Thus, we further focus specifically on the distance between the consecutive images of the embeddings.

We make a formal comparison between Positional Interpolation and Adjusted Base Frequency by analytically comparing the pairwise distances between the images given by $f^{RoPE+PI}$ and $f^{RoPE+ABF}$ for consecutive integer values of t. This corresponds to the evaluation of $q(\hat{f})$ discussed earlier. We will measure the distance between embedding images in terms of the cosine similarity metric since all versions of RoPE are norm-preserving.

$$\cos\angle(a,b) = \frac{Re\langle a,b\rangle}{\|a\|\|b\|}$$

The following result states that in a high-dimensional space, the cosine similarity $\cos\angle(f^{RoPE+ABF}(x,n+1), f^{RoPE+ABF}(x,n))$ between two consecutive embedding images of a vector x can be bounded with a value proportional to $(\log b + \log\beta)^{-1}$. Moreover, the similarity $\cos\angle(f^{RoPE+PI}(x,n+1), f^{RoPE+PI}(x,n))$ can be bounded using $\alpha(\log b)^{-1}$.

Theorem 1. *For $x \in \mathbb{R}^d$ and $n \in \mathbb{N}$, the cosine similarity between the two consecutive images of a positional embedding can be bounded as*

$\frac{\min_k x_k^2}{\|x\|^2} C_d \le \cos\angle(f(x,n+1), f(x,n)) \le \frac{\max_k x_k^2}{\|x\|^2} C_d$

where $\lim_{d\to\infty} C_d \approx \begin{cases} (\log b + \log \beta)^{-1} \textit{ if } f = f^{RoPE+ABF} \\ \alpha(\log b)^{-1} \textit{ if } f = f^{RoPE+PI} \end{cases}$ *under the assumptions of* $\alpha \ll 1$ *and* $b \gg 1$.

Proof. Let us begin the proof by writing down the expressions for the inner product between two images of RoPE variants.

$\langle f^{RoPE+PI}(x,m), f^{RoPE+PI}(x,n)\rangle = \sum_{j=0}^{\frac{d}{2}-1} \left(x_{2j}^2 + x_{2j+1}^2\right) e^{ib^{-\frac{2j}{d}}\alpha(m-n)}$

$\langle f^{RoPE+ABF}(x,m), f^{RoPE+ABF}(x,n)\rangle = \sum_{j=0}^{\frac{d}{2}-1} \left(x_{2j}^2 + x_{2j+1}^2\right) e^{ib^{-\frac{2j}{d}}\beta^{-\frac{2j}{d}}(m-n)}$

From them, we can derive the expressions for the cosine similarity between the images of the positional embeddings:

$\cos\angle(f^{RoPE+PI}(x,m), f^{RoPE+PI}(x,n)) = \frac{\sum_{j=0}^{\frac{d}{2}-1}\left(x_{2j}^2+x_{2j+1}^2\right)\sin(b^{-\frac{2j}{d}}\alpha(m-n))}{\sum_{j=0}^{d-1} x_j^2}$

$\cos\angle(f^{RoPE+ABF}(x,m), f^{RoPE+ABF}(x,n)) = \frac{\sum_{j=0}^{\frac{d}{2}-1}\left(x_{2j}^2+x_{2j+1}^2\right)\sin(b^{-\frac{2j}{d}}\beta^{-\frac{2j}{d}}(m-n))}{\sum_{j=0}^{d-1} x_j^2}$

Let's put $m = n+1$ to compare the distance between the two consecutive positional embedding images of the same vector x.

$\|x\|^2 \cos\angle(f^{RoPE+PI}(x,n+1), f^{RoPE+PI}(x,n)) = \sum_{j=0}^{\frac{d}{2}-1} \left(x_{2j}^2 + x_{2j+1}^2\right) \sin(b^{-\frac{2j}{d}}\alpha)$

$\|x\|^2 \cos\angle(f^{RoPE+ABF}(x,n+1), f^{RoPE+ABF}(x,n)) = \sum_{j=0}^{\frac{d}{2}-1} \left(x_{2j}^2 + x_{2j+1}^2\right) \sin(b^{-\frac{2j}{d}}\beta^{-\frac{2j}{d}})$
Due to the range of b, α and β that is typically considered, we can boud the arguments of the sine functions as $0 < \alpha b^{-\frac{2j}{d}} \leq 1$ as well as $0 < (\beta b)^{-\frac{2j}{d}} \leq 1$. Using that, we derive that $\sin(b^{-\frac{2j}{d}}\beta^{-\frac{2j}{d}})$ and $\sin(b^{-\frac{2j}{d}}\alpha)$ are non-negative as well as x_j^2 for any $j \in \{1, \dots d\}$. Thus, the following inequalities hold:

$$\sum_{j=0}^{\frac{d}{2}-1} \min_k x_k^2 \sin(b^{-\frac{2j}{d}}\beta^{-\frac{2j}{d}}) \leq \sum_{j=0}^{\frac{d}{2}-1} \left(x_{2j}^2 + x_{2j+1}^2\right) \sin(b^{-\frac{2j}{d}}\beta^{-\frac{2j}{d}}) \leq \sum_{j=0}^{\frac{d}{2}-1} \max_k x_k^2 \sin(b^{-\frac{2j}{d}}\beta^{-\frac{2j}{d}}),$$

$$\sum_{j=0}^{\frac{d}{2}-1} \min_k x_k^2 \sin(b^{-\frac{2j}{d}}\alpha) \leq \sum_{j=0}^{\frac{d}{2}-1} \left(x_{2j}^2 + x_{2j+1}^2\right) \sin(b^{-\frac{2j}{d}}\alpha) \leq \sum_{j=0}^{\frac{d}{2}-1} \max_k x_k^2 \sin(b^{-\frac{2j}{d}}\alpha).$$

Carrying $\min_k x_k^2$ and $\max_k x_k^2$ out of the summation signs, we obtain

$$\min_k x_k^2 \sum_{j=0}^{\frac{d}{2}-1} \sin(b^{-\frac{2j}{d}}\beta^{-\frac{2j}{d}}) \leq \sum_{j=0}^{\frac{d}{2}-1} \left(x_{2j}^2 + x_{2j+1}^2\right) \sin(b^{-\frac{2j}{d}}\beta^{-\frac{2j}{d}}) \leq \max_k x_k^2 \sum_{j=0}^{\frac{d}{2}-1} \sin(b^{-\frac{2j}{d}}\beta^{-\frac{2j}{d}}),$$

$$\min_k x_k^2 \sum_{j=0}^{\frac{d}{2}-1} \sin(b^{-\frac{2j}{d}}\alpha) \leq \sum_{j=0}^{\frac{d}{2}-1} \left(x_{2j}^2 + x_{2j+1}^2\right) \sin(b^{-\frac{2j}{d}}\alpha) \leq \max_k x_k^2 \sum_{j=0}^{\frac{d}{2}-1} \sin(b^{-\frac{2j}{d}}\alpha).$$

Introducing $C_d^{ABF} = \sum_{j=0}^{\frac{d}{2}-1} \sin(b^{-\frac{2j}{d}}\beta^{-\frac{2j}{d}})$ and $C_d^{PI} = \sum_{j=0}^{\frac{d}{2}-1} \sin(b^{-\frac{2j}{d}}\alpha)$ proves the first part of the Theorem:

$$\frac{\min_k x_k^2}{\|x\|^2} C_d^{ABF} \leq \cos\angle(f^{RoPE+ABF}(x,n+1), f^{RoPE+ABF}(x,n)) \leq \frac{\max_k x_k^2}{\|x\|^2} C_d^{ABF},$$

$$\frac{\min_k x_k^2}{\|x\|^2} C_d^{PI} \leq \cos\angle(f^{RoPE+PI}(x,n+1), f^{RoPE+PI}(x,n)) \leq \frac{\max_k x_k^2}{\|x\|^2} C_d^{PI}.$$

Now, considering the limit of C_d, we notice that due to the inequalities on the arguments of the sines, the following bounds hold:

$$(b\beta)^{-\frac{2j}{d}}\left(1-(b\beta)^{-\frac{2j}{d}}/\pi\right) \leq \sin(b^{-\frac{2j}{d}}\beta^{-\frac{2j}{d}}) \leq (b\beta)^{-\frac{2j}{d}},$$

$$\alpha b^{-\frac{2j}{d}}\left(1-\alpha b^{-\frac{2j}{d}}/\pi\right) \leq \sin(b^{-\frac{2j}{d}}\alpha) \leq \alpha b^{-\frac{2j}{d}}$$

Using the formula of geometric sums and a corollary of the exponential (second) foundational limit, we establish the limits of the sums of these bounds as $d \to \infty$:

$$\sum_{j=0}^{\frac{d}{2}-1} \alpha b^{-\frac{2j}{d}} = \frac{\alpha(b-1)b^{2/d}}{b^{2/d+1}-b} \to \alpha\frac{b-1}{b\log b} \text{ as } d \to \infty$$

$$\sum_{j=0}^{\frac{d}{2}-1} \alpha^2 b^{-\frac{4j}{d}} = \frac{\alpha^2(b^2-1)b^{4/d}}{b^{4/d+2}-b^2} \to \alpha^2\frac{b^2-1}{b^2\log b} \text{ as } d \to \infty$$

$$\sum_{j=0}^{\frac{d}{2}-1} (b\beta)^{-\frac{2j}{d}} = \frac{(b\beta-1)(b\beta)^{2/d}}{(b\beta)^{2/d+1}-b\beta} \to \frac{(b\beta)-1}{(b\beta)\log(b\beta)} \text{ as } d \to \infty$$

$$\sum_{j=0}^{\frac{d}{2}-1} (b\beta)^{-\frac{4j}{d}} = \frac{(b^2\beta^2-1)(b\beta)^{4/d}}{(b\beta)^{4/d+2}-b^2\beta^2} \to \frac{(b\beta)^2-1}{(b\beta)^2\log(b\beta)} \text{ as } d \to \infty$$

Substituting these into the bounds on $\lim_{d\to\infty} C_d$, one achieves:

$$(\log b + \log\beta)^{-1}\left(\frac{(b\beta)-1}{(b\beta)} - \frac{(b\beta)^2-1}{\pi(b\beta)^2}\right) \leq \lim_{d\to\infty} C_d^{ABF} \leq (\log b + \log\beta)^{-1}\frac{(b\beta)-1}{(b\beta)},$$

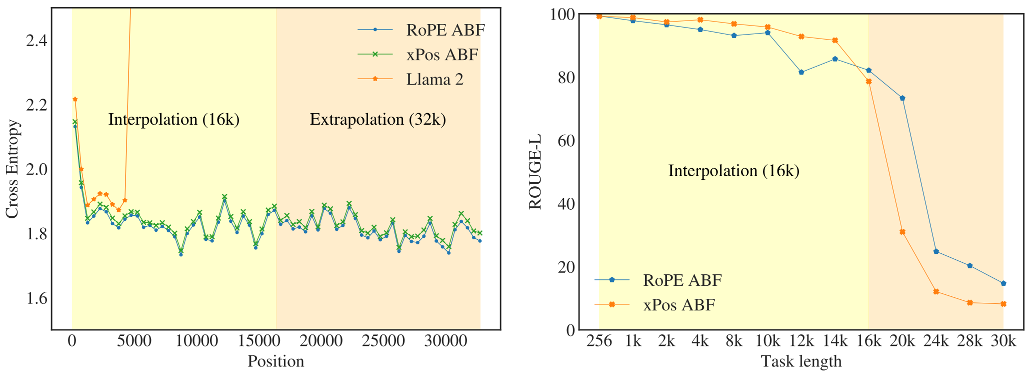
$$\alpha(\log b)^{-1}\left(\frac{b-1}{b} - \frac{\alpha}{\pi}\frac{b^2-1}{b^2}\right) \leq \lim_{d\to\infty} C_d^{PI} \leq \alpha(\log b)^{-1}\frac{b-1}{b}$$

From these bounds, one can see that in the setting considered within this paper, where $b = 10000$ and $\alpha < 1/4$, the approximation of $\lim_{d\to\infty} C_d$ used in the statement of the Theorem is of a high quality.

□

Based on this theoretical derivation, we return to the interpretation of our experimental resuts. On one hand, the experiments have shown that the model can adapt to the new sequence length with both RoPE PI ($\alpha = 1/4$ or $\alpha = 1/8$) and RoPE ABF ($\beta = 50$). Thus, we can conclude that the chosen hyperparameters provide a sufficient degree of approximation of RoPE images under $b = 10000$. In other words, both $d(f, f^{RoPE+ABF})$ and $d(f, f^{RoPE+PI})$ are small enough to allow rapid adaptation. On the other hand, comparing the expressions of C_d for RoPE ABF and RoPE PI, we can observe that for the values of $\alpha = \frac{1}{4}$ or $\alpha = \frac{1}{8}$ and $b = 10000$ that were used in our experiments, the granularity (the distance between two consecutive images of RoPE) is much lower for the RoPE PI ($\alpha(\log b)^{-1} \approx 0.027$) than for RoPE ABF ($(\log b + \log\beta)^{-1} \approx 0.076$) with $\beta = 50$. We further hypothesise that the higher degree of granularity is related to the higher evaluation on the downstream tasks of the RoPE ABF variant compared to RoPE PI because it makes the task of distinguishing between the positional embedding images simpler for the model. In other words, this corresponds to the case of $q(f^{RoPE+ABF}) > q(f^{RoPE+PI})$.

Throughout this consideration we implicitly assumed that the distance between the consecutive images of an embedding is smaller than the distance between any other pair of the images. While this assumption is likely to hold true in a high-dimensional space, significantly increasing the parameter of β in RoPE ABF may violate this assumption due to the changed geometry of the embedding curve.



(a) Validation loss calculated at each position of 32,768 context window.

(b) Context window probing with FIRST-SENTENCE-RETRIEVAL task.

Figure 9: Evaluation on our 70B model's extrapolation capabilities.

C Length Extrapolation Results

Despite not the focus of this work, extrapolation is an important property for long context models. Extrapolation refers to a model's ability to conduct inference on input sequences that are longer than its training sequences. We evaluate how our 70B model extrapolates with two tasks:

- **Validation loss at each position**: In Figure 9a, we visualize the average loss at each position of the 32,768 sequence length where the first 16,384 is the interpolation area (within training sequence length) and the second half is extrapolation. We use 50 batches of samples and average across them. To make plots smoother, we also take the mean of losses every 500 positions. As we can see, our 70B model with either ROPE ABF or XPOS ABF maintain the loss in the extrapolation area. To contrast this, we also plot the result for LLAMA 2 with 4,096 context window: the loss explodes after the position goes beyond training sequence length, which suggests that LLAMA 2 does not extrapolate effectively.
- **Synthetic FIRST-SENTENCE-RETRIEVAL task**: To complement validation loss evaluation, we also test our 70B model with two different PEs on the context probing task. Unlike validation loss task where it is hard to find data samples that require very long range dependencies consistently, FIRST-SENTENCE-RETRIEVAL imposes a very strict requirement for models to attend with a specific length. In Figure 9b, we visualize the results up to 32,768 where we do see some performance degradation when the model needs to extrapolate. In addition, we observe that, despite often considered as having better extrapolation properties, XPOS ABF does not outperform ROPE ABF in our setting.

D Self-Instruct Data

As described in Section 4.3, we use LLAMA 2 CHAT to bootstrap self-instruct data for instruct finetuning. In this section we describe the detailed procedure as well as providing the necessary prompts used for generating this dataset. The main challenge is that we need an automated process of generating long context instruct data with only short context models at hand. The core idea behind this is to split the long documents into chunks of texts that can fit into short model's context and apply self-instruct. We focus primarily on question answering dataset. We first split the long document into smaller chunks, and for each chunk we construct a prompt as in Figure 10 which gets fed into LLAMA 2 CHAT to get a question-answer pair. To diversify the question types, we randomly choose between the two prompts that ask for either normal or short answers. Once we extract the question and answer from the response (using tags as required by the prompt), we can construct long question answering instruct data together with the original long document, using the templates in Figure 11 of the corresponding answer type.

Normal Answer Prompt:

```
[INST] You are given a text chunk (delimited by triple quotes) taken from a long
text. Write a question about this text and provide the correct answer. The answer
needs to be based on the text. This question will later be used as a reading
comprehension test over the entire document. Wrap the question and answer using
XML tags (<question> and </question>, <answer> and </answer>).
"""
{TEXT_CHUNK}
"""
[/INST]
```

Short Answer Prompt:

```
[INST] You are given a text chunk (delimited by triple quotes) from a long
document. Based on information from the text, come up with a specific question
**which can be answered in a few words or a single phrase** and provide the
correct answer without explanation. The answer needs to be based on the text.
This question will later be used as a reading comprehension test over the
entire document. Wrap the question and answer using XML tags (<question>
and </question>, <answer> and </answer>). Again, the answer needs to be short.
"""
{TEXT_CHUNK}
"""
[/INST]
```

Figure 10: Prompts used for generating question and answer pairs by boostrapping LLAMA 2 CHAT. We split the long documents into chunks and feed each chunk into one of the prompts with equal probability. We prompt the models to wrap the answer with XML tags, which enables more accurate answer extraction.

Normal Answer Data Template:

```
[INST] You are given a long text (delimited by triple quotes) and a question.
Read the text and answer the question in the end.
"""
{FULL_DOCUMENT}
"""
Question: {QUESTION}
[/INST]
{ANSWER}
```

Short Answer Data Template:

```
[INST] You are given a long text (delimited by triple quotes) and a question.
Read the text and answer the question in the end as concisely as you can,
using a single phrase or sentence if possible. Do not provide any explanation.
"""
{FULL_DOCUMENT}
"""
Question: {QUESTION}
[/INST]
{ANSWER}
```

Figure 11: Data templates for constructing long question-answer data. The question and answer pair is extracted from the response of LLAMA 2 CHAT.