CD-Pos: Long Context Generalization in LLMs Through Continuous and Discrete Position Synthesis

Zhiyuan Hu [*1] **Yuliang Liu** [*2] **Jinman Zhao** [3] **Suyuchen Wang** [4 5]
Yan Wang [6] **Wei Shen** [7] **Chao Yin** [2] **Bryan Hooi** [1]

Abstract

Large language models (LLMs) are critical for natural language processing and multi-modal tasks, but face challenges in tasks requiring long context windows due to computational and memory limitations. Existing methods to extend these windows are resource intensive. The proposed Continuous and Discrete Position Synthesis (CD-Pos) addresses these issues by using synthesized position indices to expand context windows efficiently. CD-Pos divides sequences into segments with continuous indices, enhancing token distance and preserving local information. Empirical evaluations show that CD-Pos effectively extends context windows up to 128k while maintaining LLMs' performance in general tasks.

1. Introduction

LLMs are crucial for NLP and multi-modal tasks. However, they face challenges in applications like in-context learning (Brown et al., 2020), long document summarization (Koh et al., 2022), long-form QA (Krishna et al., 2021), and document-level retrieval (Callan, 1994). These challenges stem from the limited effective context window size during the pretraining process, posing new challenges in generalizing over long contexts.

A straightforward approach is to continually pre-train or fine-tune these models on extensive texts (Fu et al., 2024). However, expanding the context window usually results in a quadratic increase in computational and memory costs. According to the training setup in (Fu et al., 2024), extending the LlaMA-2 7B model's context window from 4k to 80k using 8 A100 GPUs (80G each) takes five days. The resource and time costs increase significantly for larger models and longer training periods. In addition to the methods mentioned, there are techniques aimed at extending the context window length more efficiently during fine-tuning, including PI (Chen et al., 2023), Yarn (Peng et al., 2024), and LongLoRA (Chen et al., 2024). However, these techniques still require full-length fine-tuning, meaning they must fine-tune with the context of the target length, which is both memory- and time-intensive. Meanwhile, the Randomized Positional Encoding Scheme (Ruoss et al., 2023) and PoSE (Zhu et al., 2023) simulate longer inputs within a fixed window by adjusting position indices, enabling LLMs which are trained on shorter contexts but can be extended to longer context windows. However, randomized position embeddings in (Ruoss et al., 2023) disrupt local sentence structures by exaggerating the dependency lengths between neighboring tokens. PoSE, on the other hand, only considers two chunks to mimic the position index, consistently omitting longer dependencies in the sequence. This distortion creates a significant generalization gap in understanding token relationships across the sequence when extending LLMs to a long context window.

To address the aforementioned issues, we introduce **C**ontinuous and **D**iscrete **Pos**ition Synthesis (CD-Pos), a method designed to utilize short text with synthesized position indices to expand the effective context window of LLMs through continual pre-training. CD-Pos constructs positional indices to mimic long inputs. As illustrated in Figure 1, we divide the sequence into various segments, which can be either sentences or paragraphs. Within each segment, the positional indices are continuous, but there are positional gaps between different segments. This approach offers two main advantages: **(1) Increased Distance Among Tokens.** CD-Pos allows training samples to have a greater distance, which is essential for LLMs to recognize long-range dependencies within a short sequence. **(2) Conservation of Local Information.** To have continuous sequences is crucial for LLMs as it can maintain LLMs' awareness of the local dependency structure. We also present elaboration and derivation for these two points in Section 6.

Our empirical evaluation on Needle In A Haystack (gkam-

[*]Equal contribution [1]National University of Singapore, Singapore [2]Nanjing University, China [3]University of Toronto, Canada [4]Mila, Québec AI Institute, Canada [5]Université de Montréal, Canada [6]Tencent Inc, China [7]Baidu Inc, China. Correspondence to: Zhiyuan Hu <zhiyuan_hu@u.nus.edu>.

Accepted by the Workshop on Long-Context Foundation Models (LCFM) at ICML 2024,

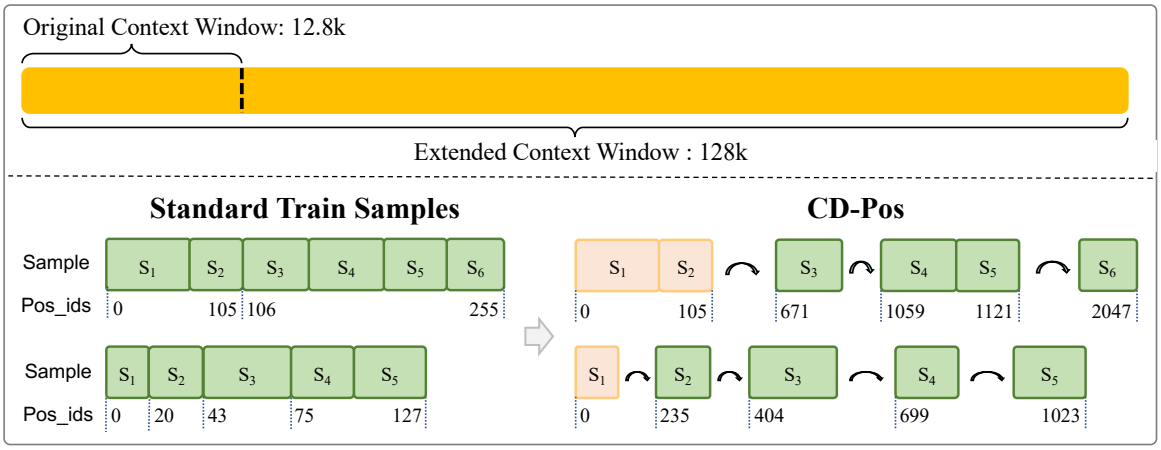

Figure 1. The figure demonstrates that the goal of CD-Pos is to extend the context window from a short scale (e.g., Original Context Window: 12.8k) to a long scale (e.g., Extended Context Window: 128k, as shown at the top). We use two examples to illustrate how CD-Pos divides the sequence into various segments and simulates positional indices into continuous and discrete segments. Each is initially positioned within a 256-token and 128-token, and independently applies our method to stretch lengths to 2048 and 1024 separately.

radt, 2023) and RULER (Hsieh et al., 2024) validate the effectiveness of CD-Pos from the length of 20k to 128k. Additionally, we test our method's performance in general tasks including GSM8K (Cobbe et al., 2021), MMLU (Hendrycks et al., 2021), PiQA (Bisk et al., 2019) and HumanEval (Chen et al., 2021) to estimate whether our method affects LLMs' original abilities in general domains, showing that our method keeps LLMs almost at their original foundational abilities.

To summarize, our contributions are as follows:

- We propose an efficient training method, **C**ontinuous and **D**iscrete **Pos**ition Synthesis (CD-Pos), aimed at using synthesized position indices to expand the effective context window.
- We assess CD-Pos's effectiveness in enhancing distances while preserving local information by measuring the average distance among tokens and the average continual length of segments.
- Empirical experiments conducted on context lengths from 20k to 128k of LlaMa-2-7B and LlaMa3-8B validate the effectiveness of our proposed method.

2. Preliminary

The approach that is widely used in previous pre-trained language models such as BERT (Devlin et al., 2018) is to add position embedding vectors to word embedding vectors directly. For a sequence of tokens represented as $w_1, w_2, \cdots, w_L$, with their corresponding embeddings $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_L$, let $\mathbf{p}_1, \mathbf{p}_2, ...\mathbf{p}_L$ be absolute position embedding, the position encoding of query($\mathbf{q}$) and key($\mathbf{k}$) are $\mathbf{q}_m = W_q(\mathbf{x}_m + \mathbf{p}_m)$ and $\mathbf{k}_n = W_k(\mathbf{x}_n + \mathbf{p}_n)$. Then the unnormalized attention scores are calculated by dot-producting two vectors: $score(\mathbf{q}_m, \mathbf{k}_n) = \mathbf{q}_m^T \cdot \mathbf{k}_n$.

Rotary Position Embedding (RoPE) (Su et al., 2024) is proposed to integrate relative positional information by modulating the query and key vectors in the attention mechanism. Let D denote the dimension of hidden layers, the transformations applied are as follows:

$$\mathbf{q}_m = W_q \mathbf{x}_m \cdot e^{im\theta}, \quad \mathbf{k}_n = W_k \mathbf{x}_n \cdot e^{in\theta}, \qquad (1)$$

where W_q and W_k are $|D| \times |D|$ projection matrices, m and n are the positions of the tokens, and θ is a constant that adjusts the rotation based on token positions.

$$\theta_i = 10000^{\frac{-2i}{D}}$$

RoPE operation on $\vec{q} = W_q \mathbf{x}_m$ results $\mathbf{q}_m =$:

$$\begin{bmatrix} q_0 \\ q_1 \\ \vdots \\ q_{D-2} \\ q_{D-1} \end{bmatrix} \otimes \begin{bmatrix} \cos m\theta_0 \\ \cos m\theta_0 \\ \vdots \\ \cos m\theta_{\frac{D}{2}-1} \\ \cos m\theta_{\frac{D}{2}-1} \end{bmatrix} + \begin{bmatrix} q_1 \\ q_0 \\ \vdots \\ q_{D-1} \\ q_{D-2} \end{bmatrix} \otimes \begin{bmatrix} -\sin m\theta_0 \\ \sin m\theta_0 \\ \vdots \\ -\sin m\theta_{\frac{D}{2}-1} \\ \sin m\theta_{\frac{D}{2}-1} \end{bmatrix}$$

The real part of the inner product between $\mathbf{q}_m$ and $\mathbf{k}_n$ captures the relative positional information, facilitating the model's understanding of token distances.

3. Related Works

Position Encoding Various position encoding methods have been proposed to perform extrapolation such as ALiBi (Press et al., 2022), xPos (Sun et al., 2023) and KERPLE (Chi et al., 2022). RoPE (Su et al., 2024), the most widely used one, introduces a more complex mechanism.

Efficient Pretraining or Fine-tuning Methods Position Interpolation (PI)(Chen et al., 2023) downsizes position indices of long text to the original window size. NTK Interpolation(Peng & Quesnelle, 2023) adjusts rotation speed for small positions and linear interpolation for large ones. YaRN (Peng et al., 2024) improves NTK Interpolation with NTK-by-parts scaling to accommodate different RoPE features. LM-Infinite (Han et al., 2024) encodes absolute positions for starter tokens and masks middle tokens, retaining relative positions for rare tokens. Randomized Positional Embedding (Ruoss et al., 2023) simulates long text input with shorter texts by randomly selecting position indices. PoSE (Zhu et al., 2023) uses a fixed context window, dividing it into chunks with skipping bias terms, enabling adaptation to all positions within the target length. LongLoRA (Chen et al., 2024) replaces ordinary attention with shift short attention. Temp Lora (Wang et al., 2024) integrates context details into a temporary Lora module, incrementally trained with previously generated text.

4. Methodology

We aim to utilize the current data with context window L to enable the model with larger input context length $\hat{L}$ by further continual pre-training in the data with synthesized position indices. Let S be the original sequence. We define a function **seg** : $S \rightarrow \{s_1, s_2, \ldots, s_N\}$ that partitions S into N segments, where each segment s_i can be either a sentence or a paragraph, for $1 \leq i \leq N$. The function **seg** satisfies the following conditions:

$$S = s_1 \cup s_2 \cup \cdots \cup s_N \tag{2}$$

The union of all segments reconstructs the original sequence and segments are disjoint:

$$s_i \cap s_j = \emptyset \quad \text{for all} \quad i \neq j \tag{3}$$

To vary the spacing between each segment, we will randomly skip some position indices from 0 to M, where M is a parameter of our method. When $M = 0$, the position indices of the two segments will be continuous.

We start by defining $\mathbf{pos}(s_i)$ as the position index of the first token of segment s_i. For each segment, the position indices are sequentially increased by 1 for each token within that segment. The position index of the first token in the first segment is set to 1, i.e., $\mathbf{pos}(s_1) = 1$.

For subsequent segments, we introduce a random skip represented by a function $g(s_i)$ which takes values from $0, 1, \ldots, M$. This function represents the gap before the start of segment s_i and is determined randomly for each segment. Thus, the position index of the first token of segment s_i, for $i \geq 2$, can be defined recursively as follows:

$$\text{pos}(s_i) = \text{pos}(s_{i-1}) + |s_{i-1}| + g(s_i) \tag{4}$$

Where $|s_{i-1}|$ represents the number of tokens in segment s_{i-1}. We repeat this process until the position index of the last token of the last segment s_N does not exceed $\hat{L}$.

To achieve comprehensive coverage of the target context window, we re-sample both the length and skipping term of every chunk for each training example.

5. Experimental Setup

5.1. Baselines

Randomized Positional Encoding Scheme (RPES) (Ruoss et al., 2023) simulates the positions of longer sequences and randomly selects an ordered subset to match longer length.

Positional Skip-wisE (PoSE) (Zhu et al., 2023) simulates long inputs using a fixed context window. It divides the original context window into two chunks and applies distinct skipping bias terms to manipulate the position indices of each chunk. These bias terms and the lengths of the chunks are changed for each training example, enabling the model to adapt to all positions within the target length.

5.2. Evaluation

Benchmarks of Long Context Generalization The **Needle In A Haystack (NIAH)** framework (gkamradt, 2023) tests LLMs' ability to retrieve hidden information by embedding a "needle" (fact) within a "haystack" (long document). **RULER** (Hsieh et al., 2024) offers flexible sequence lengths and task complexities with 13 sub-task categories, including retrieval and question answering. **LongBench** (Bai et al., 2023) is the first bilingual benchmark for long context understanding, featuring 21 tasks in six categories. **LooGLE** (Li et al., 2023) evaluates long context understanding with post-2022 documents (24,000+ tokens) and 6,000 questions, including 1,100 cross-validated question-answer pairs.

Datasets for Assessment of Fundamental Abilities of LLMs We use three benchmarks to test if the continual pre-training process affects LLMs' fundamental abilities within their original context length. **MMLU** includes many academic subjects like mathematics, philosophy, law and medicine (Hendrycks et al., 2021). **GSM8K** (Cobbe et al., 2021) is a benchmark of math problems. **HumanEval** (Chen et al., 2021) is a code problem solving dataset.

5.3. Setup

We utilize the sample with 30% length of extended context window to train the LLMs and leverage FlashAttention 2 (Dao et al., 2022) and DeepSpeed Zero 3 (Aminabadi et al., 2022) to enhance the efficiency of training and optimize the GPU memory demands and we use LightSeq (Li et al., 2024) to train Llama-3-8B. Further details including RoPE scaling, Batch Size, Hours to Train and others are in Appendix A.

Table 1. Performance of different methods in context generalization benchmarks and fundamental abilities benchmarks

Model	Length	Method	NIAH	RULER	LongBench	Loogle	MMLU	GSM8K	HumanEval
Llama2-7B	4k	Base Model	-	-	-	-	**44.4**	**12.2**	**9.4**
	20k	RPES	91.3	50.3	17.2	75.9	38.0	8.9	10.0
		PoSE	81.2	57.3	16.2	72.2	40.9	8.5	7.5
		CD-Pos	**96.0**	**57.5**	**18.5**	**76.9**	40.0	9.1	4.9
	80k	RPES	61.3	17.4	17.2	69.8	32.6	4.7	4.3
		PoSE	62.4	43.3	17.6	74.6	39.7	7.7	10.8
		CD-Pos	**75.9**	**43.6**	**18.1**	**75.2**	39.4	10.0	7.3
	128k	RPES	54.6	8.2	17.4	74.8	32.6	4.2	4.7
		PoSE	57.3	18.4	**18.2**	74.0	39.2	8.5	8.1
		CD-Pos	**75.0**	**22.7**	17.7	**79.0**	38.8	9.2	4.9
Llama3-8B	8k	Base Model	-	-	-	-	**65.7**	**71.4**	**37.5**
	80k	RPES	99.2	69.6	20.2	74.6	57.3	40.9	5.9
		PoSE	**100.0**	73.2	19.9	74.5	60.0	45.9	9.0
		CD-Pos	90.7	**75.4**	**20.4**	**74.8**	59.7	46.6	11.2
	128k	RPES	61.6	22.5	**26.7**	72.1	56.1	36.0	5.1
		PoSE	94.8	64.3	22.7	74.4	58.9	40.2	7.5
		CD-Pos	**100.0**	**68.5**	23.5	**74.6**	59.7	44.2	9.5

6. Performance and Analysis

Long Context Generalization The CD-Pos method shows an average improvement of 13.9% over RPES and 8.4% over PoSE in the NIAH task. In the RULER evaluation, CD-Pos outperforms RPES by 11.6% and PoSE by 2.2%. In the LongBench evaluation, CD-Pos outperforms PoSE by 0.7%. In the Loogle evaluation, CD-Pos outperforms RPES by 2.7% and PoSE by 2.3%. Especially, In the NIAH task, Llama2-7B (80k) shows a 20.6% improvement with CD-Pos over PoSE. In the RULER task, Llama3-8B (128k) improves by 4.2%.

Fundamental Abilities Maintainment Table 1 shows that LLMs can nearly maintain their original abilities with short inputs, as seen by the minor performance drop in MMLU, which covers 57 subjects across STEM. However, their math and coding abilities decrease significantly in GSM8K and HumanEval. To improve these areas, incorporating math and coding data in the continual pre-training process should be further considered.

Distance Among Tokens and Continual Length of Segments To evaluate the effectiveness of CD-Pos in improving distances while maintaining local information, we calculated the average distance among tokens and the average continuous segment length for different methods. As shown in Figure 2, the CD-Pos approach achieves approximately twice the token distance compared to PoSE in 128k setting. Additionally, CD-Pos maintains an average continuous segment length of 88, which helps the LLM recognize local dependency structures. In contrast, the average continuous segment length with RPES is nearly 0, which greatly disrupts local sentence structures.

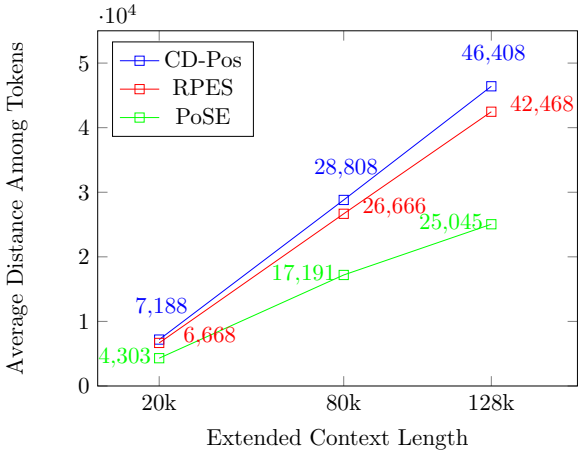

Figure 2. Comparison of average distance among tokens for different methods and various context window settings.

7. Conclusion

Our CD-Pos method tackles the challenge of long context generalization by generating position indices to extend context windows effectively. By breaking down sequences into segments with continuous indices, CD-Pos enlarges token distance while maintaining local information. This significantly enhances the ability of LLMs to handle longer contexts. Our tests show that CD-Pos can increase context windows to up to 128k tokens. Comparative studies demonstrate that CD-Pos outperforms RPES and PoSE by 14.7% and 3.7%, respectively, in the Needle In A Haystack task and RULER evaluation. Moreover, the LLMs retain their performance with short inputs, and additional pre-training on mathematical and coding data may be beneficial.

References

Aminabadi, R. Y., Rajbhandari, S., Awan, A. A., Li, C., Li, D., Zheng, E., Ruwase, O., Smith, S., Zhang, M., Rasley, J., et al. Deepspeed-inference: enabling efficient inference of transformer models at unprecedented scale. In *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–15. IEEE, 2022.

Bai, Y., Lv, X., Zhang, J., Lyu, H., Tang, J., Huang, Z., Du, Z., Liu, X., Zeng, A., Hou, L., et al. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*, 2023.

Bisk, Y., Zellers, R., Bras, R. L., Gao, J., and Choi, Y. PIQA: reasoning about physical commonsense in natural language. *CoRR*, abs/1911.11641, 2019. URL http://arxiv.org/abs/1911.11641.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Callan, J. P. Passage-level evidence in document retrieval. In *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*, pp. 302–310. Springer, 1994.

Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A. N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., and Zaremba, W. Evaluating large language models trained on code, 2021.

Chen, S., Wong, S., Chen, L., and Tian, Y. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023.

Chen, Y., Qian, S., Tang, H., Lai, X., Liu, Z., Han, S., and Jia, J. LongloRA: Efficient fine-tuning of long-context large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=6PmJoRfdaK.

Chi, T.-C., Fan, T.-H., Ramadge, P. J., and Rudnicky, A. Kerple: Kernelized relative positional embedding for length extrapolation. *Advances in Neural Information Processing Systems*, 35:8386–8399, 2022.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

Dao, T., Fu, D., Ermon, S., Rudra, A., and Ré, C. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Fu, Y., Panda, R., Niu, X., Yue, X., Hajishirzi, H., Kim, Y., and Peng, H. Data engineering for scaling language models to 128k context. *arXiv preprint arXiv:2402.10171*, 2024.

gkamradt. Llmtest_needleinahaystack: Doing simple retrieval from llm models. https://github.com/gkamradt/LLMTest_NeedleInAHaystack/tree/main, 2023. [Online; accessed 29-December-2023].

Han, C., Wang, Q., Peng, H., Xiong, W., Chen, Y., Ji, H., and Wang, S. Lm-infinite: Zero-shot extreme length generalization for large language models, 2024.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding, 2021.

Hsieh, C.-P., Sun, S., Kriman, S., Acharya, S., Rekesh, D., Jia, F., and Ginsburg, B. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

Koh, H. Y., Ju, J., Liu, M., and Pan, S. An empirical survey on long document summarization: Datasets, models, and metrics. *ACM computing surveys*, 55(8):1–35, 2022.

Krishna, K., Roy, A., and Iyyer, M. Hurdles to progress in long-form question answering. *arXiv preprint arXiv:2103.06332*, 2021.

Li, D., Shao, R., Xie, A., Xing, E. P., Ma, X., Stoica, I., Gonzalez, J. E., and Zhang, H. Distflashattn: Distributed memory-efficient attention for long-context llms training, 2024.

Li, J., Wang, M., Zheng, Z., and Zhang, M. Loogle: Can long-context language models understand long contexts? *arXiv preprint arXiv:2311.04939*, 2023.

Peng, B. and Quesnelle, J. 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, 2023.

Peng, B., Quesnelle, J., Fan, H., and Shippole, E. YaRN: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=wHBfxhZu1u.

Press, O., Smith, N., and Lewis, M. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=R8sQPpGCv0.

Ruoss, A., Delétang, G., Genewein, T., Grau-Moya, J., Csordás, R., Bennani, M., Legg, S., and Veness, J. Randomized positional encodings boost length generalization of transformers. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1889–1903, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.161. URL https://aclanthology.org/2023.acl-short.161.

Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2023.127063. URL https://www.sciencedirect.com/science/article/pii/S0925231223011864.

Sun, Y., Dong, L., Patra, B., Ma, S., Huang, S., Benhaim, A., Chaudhary, V., Song, X., and Wei, F. A length-extrapolatable transformer. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14590–14604, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.816. URL https://aclanthology.org/2023.acl-long.816.

Wang, Y., Ma, D., and Cai, D. With greater text comes greater necessity: Inference-time training helps long text generation, 2024.

Zhu, D., Yang, N., Wang, L., Song, Y., Wu, W., Wei, F., and Li, S. Pose: Efficient context window extension of llms via positional skip-wise training. *arXiv preprint arXiv:2309.10400*, 2023.

A. Experimental Setups

Elaboration of experimental setup for different model and context window settings.

Model	**Llama2-7B**			**Llama3-8B**	
Extended Context Length	20k	80k	128k	80k	128k
Training Sample Length	6k	24k	38.4k	24k	38.4k
RoPE scaling	Dynamic NTK	Dynamic NTK	Dynamic NTK	Dynamic NTK	Dynamic NTK
Batch Size	96	96	96	96	96
Steps	104	104	104	104	104
Total Tokens	60M	240M	384M	240M	384M
Learning Rate	5e-5	5e-5	5e-5	5e-5	5e-5
# GPUs and Type	1×A800/H100	1×A800/H100	1×A800/H100	2×A800/H100	2×A800/H100
Hours to Train	7/4.5	30/18	45/27	25/16	48/29

Table 2. Training Details