LongAttn: Selecting Long-context Training Data via Token-level Attention

Anonymous ACL submission

Abstract

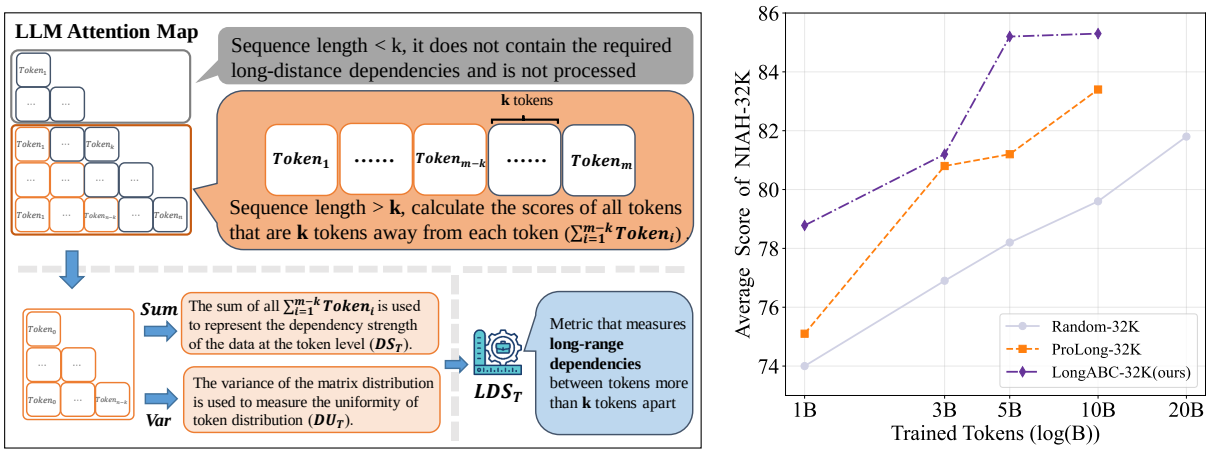
With the development of large language models (LLMs), there has been an increasing need for significant advancements in handling long contexts. To enhance long-context capabilities, constructing high-quality training data with **long-range dependencies** is crucial. Existing methods to select long-context data often rely on sentence-level analysis, which can be greatly optimized in both performance and efficiency. In this paper, we propose a novel token-level framework, **LongAttn**, which leverages the self-attention mechanism of LLMs to measure the long-range dependencies for the data. By calculating token-level dependency strength and distribution uniformity of token scores, LongAttn effectively quantifies long-range dependencies, enabling more accurate and efficient data selection. We filter **LongABC-32K** from open-source long-context datasets (ArXiv, Book, and Code). Through our comprehensive experiments, LongAttn has demonstrated its excellent **effectiveness**, **scalability**, and **efficiency**. We will release our code and the high-quality long-context training data LongABC-32K upon acceptance.

1 Introduction

Large language models (LLMs) have achieved impressive performance across a broad spectrum of traditional natural language processing tasks (Touvron et al., 2023). To effectively address real-world applications, these models further require enhanced capabilities in handling longer contexts, particularly in key areas such as in-context learning (Brown et al., 2020), real-world question-answering based on lengthy documents (Wang et al., 2024b), long-context dialogue with historical context (Packer et al., 2023), and comprehensive document summarization (Koh et al., 2022).

To enhance LLMs' long-context processing capabilities, data engineering remains fundamental. Simple methods to construct long-context datasets are through naive methods like concatenating short texts or randomly sampling existing sources (e.g., CommonCrawl, GitHub). However, studies by de Vries (2023) and Chen et al. (2024a) emphasize that data obtained through such approaches fail to effectively improve long-context capabilities of LLMs because the data lack meaningful long-range dependencies. Inspired by this, a line of studies focus on exploring the identification and selection of high-quality long-context with consideration of relations between text segments were proposed. ProLong (Chen et al., 2024a) measures long-range dependencies between segments based on the relative perplexity and relative distance. Lv et al. (2024) develop a set of metrics including complexity, coherence, and cohesion based on various kinds of text segments (i.e., sliding windows, sentences, paragraphs) to measure the quality of long texts. However, these methods have two main drawbacks: (1) Linguistic metrics do not fully align with the underlying mechanisms of LLMs, as they often fail to capture fine-grained token-level relationships. (2) They are computationally expensive and inefficient. For example, ProLong reports that the speed of a 7B parameter model is roughly 1/16 of that of a 350M parameter model, making such methods challenging to scale for LLMs.

Attention mechanisms have been proven to effectively model context understanding (Beltagy et al., 2020; Zaheer et al., 2020). Some studies focusing on attention mechanisms and positional encoding have shown that they can significantly improve a model's long-context ability (Peng and Quesnelle, 2023; Peng et al., 2023). Motivated by this, we propose to address the limitations of sentence-level selection methods leveraging the rich information provided in the attention mechanism. Specifically, we propose LongAttn, a simple yet effective framework that leverages the attention patterns of LLMs to analyse token-level dependency for long-context data selection.



(a) (b)

Figure 1: **(a)** How to measure long-range dependencies at the token level by using the self-attention mechanism. DS_T indicates that the tokens in this data have strong long-distance dependencies, while DU_T prevents negative impacts from individual tokens' high scores. **(b)** The comparison of long-context retrieval capabilities of models trained with different scales of tokens selected randomly, with sentence-level ProLong, and with LongAttn (ours).

LongAttn utilizes the long-range dependency indicator, LSD_T, to measure the strength of dependencies between tokens separated by a distance of at least k, which we define as **the minimum token distance**. We break down the indicator into two scores: dependency strength(DS_T) and distribution uniformity(DU_T). As shown in Figure 1a, DS_T measures the strength of dependencies between tokens separated by a distance of at least k, and DU_T serves as a correction term, ensuring a consistent distribution of token scores and preventing individual tokens with excessively high attention scores from skewing the overall dependency assessment. To enhance computational efficiency and avoid the **Attention Sink** (Xiao et al., 2023), we use the attention score calculated by the first decoder layer of LLaMA (Dubey et al., 2024). To better integrate DS_T and DU_T, we normalize them to the same value range and then multiply the distribution uniformity (DU_T) by a correction factor α. In this way, our framework effectively quantifies the degree of contextual information aggregation at the token level, providing a reliable criterion for selecting high-quality long-context data.

Through comprehensive experiments, the LongAttn framework has demonstrated significant advantages. We selected **A**rxiv, **B**ook, and **C**ode as the long-context datasets to be studied. After pre-processing, we used the LongAttn framework to make selections, and the resulting data is referred to LongABC-32K. Datasets selected using the ProLong framework (Chen et al., 2024a) and random selection mechanism are designated as ProLong-32K and Random-32K, respectively. As shown in Figure 1b, we compare the long-context retrieval abilities of models trained on these datasets across different token scales. The experimental results demonstrate that models trained on LongABC-32K consistently perform the best, even surpassing those trained on 20B tokens from the randomly selected dataset, despite using only 5B tokens. Through further experiments, we found that, in addition to its **effectiveness** (As seen in 5), LongAttn exhibits excellent **scalability** (Performs better with attention map from larger models, as seen in 6.2) and **efficiency** (As seen in 6.3). Our contributions are summarized as follows:

- We propose LongAttn, a framework which is the first to analyze long-range dependencies at the token level by using self-attention mechanisms.
- We will release LongABC-32K, a high-quality long-context dataset with strong long-range dependencies.
- Through comprehensive experiments, we have demonstrated LongAttn's excellent effectiveness, scalability, and efficiency.

2 Related Work

Long-context LLMs The ability to process extensive contextual information is a crucial aspect of language models, with context length serving as a key determinant of their processing capacity. To enable models to accommodate longer inputs and outputs, numerous methodologies have been de-

veloped to modify the model's architecture. Some approaches focus on altering positional encoding techniques. For instance, Chen et al. (2023a) introduced Positional Interpolation, while PoSE (Zhu et al., 2023) simulates longer texts by modifying positional encodings during training. Similar methods include YaRN (Peng et al., 2023) and LongRoPE (Ding et al., 2024). Other strategies involve modifying the attention mechanism. For example, STRING (An et al., 2024) shifts well-trained positions to overwrite originally ineffective positions during inference, and SelfExtend (Jin et al., 2024) extends the context window of Large Language Models (LLMs) by constructing bi-level attention information. Besides the methods used, the training data is also crucial. Below are related works on data.

Pre-training data Training data that exhibits long-range dependency patterns is crucial for enhancing the model's ability to handle extended contextual information. For post-training data, numerous methodologies have been explored to generate synthetic long-context data (Wang et al., 2024a; Chen et al., 2024b; Bai et al., 2024; Wu et al., 2024). Conversely, for pre-training data, the predominant approach involves the curation and selection of relevant text from existing corpora, which is exemplified by prominent models including Qwen (Bai et al., 2023a) and LLaMA (Touvron et al., 2023). While scaling laws suggest that a model's capabilities improve with more data (Kaplan et al., 2020), large volumes of data bring about high resource demands. Therefore, optimizing data utilization more effectively should become a key area of research. ProLong (Chen et al., 2024a) proposes a framework for calculating **long-distance dependencies** of data at the sentence level. LongWanjuan (Lv et al., 2024) also designed metrics and filtered data based at the sentence level. However, Xiong et al. (2023) assert that the key factor affecting the long-context ability of LLMs is the positional encoding's capacity to aggregate information from distant tokens. Our method focuses on token-level long-distance dependencies to select high-quality long-context data.

3 Methodology

As shown in Figure 2, our proposed method can be divided into three steps. Firstly, we gather and preprocess the data to a predetermined length. Subsequently, we employ the self-attention mechanism of a LLM to compute the long-distance dependency score for each data instance. Finally, we filter the data based on the score and utilize the refined dataset for continued pre-training of the model.

3.1 Data Collection and Preprocessing

To ensure the training data is suitable for long-context modeling, we carefully curate and preprocess our dataset. We choose books, code, and Arxiv papers as our primary sources of long-context data, drawing from open-source pre-training datasets such as RedPajama (Weber et al., 2024) and Dolma (Soldaini et al., 2024). These sources are known for their rich content and long sequences, which are essential for training models with extended context windows.

Given that the computational complexity of self-attention layers grows quadratically with sequence length, we set the context length to 32k tokens in this work. This length strikes a balance between capturing long-range dependencies and maintaining reasonable computational complexity. To segment/divide the data into 32k-token chunks/segments, we employ a sliding-window approach, which is more effective than naive truncation in preserving the integrity of the information. Let the total number of tokens in a text be n. The sliding-window strategy is as follows:

- If $32768(32k) < n \leq 65536(64k)$, take both the front and back windows.
- If $65536(64k) < n \leq 98304(96k)$, take the front, back, and middle windows.
- If $n > 98304(96k)$, iteratively take the front and back windows until one of the two conditions above is met.

The detailed algorithm is presented in Appendix A. After preprocessing, we obtain the long-context pre-training dataset **LongABC-32K-Raw**, which we denote as $\mathcal{D}$.

3.2 Assess Long-distance Dependency via Token-level Attention

To effectively select high-quality long-context data, we need to accurately measure the long-range dependencies within the data. In this section, we detail the process of assessing long-distance dependencies in the data using token-level attention mechanisms.

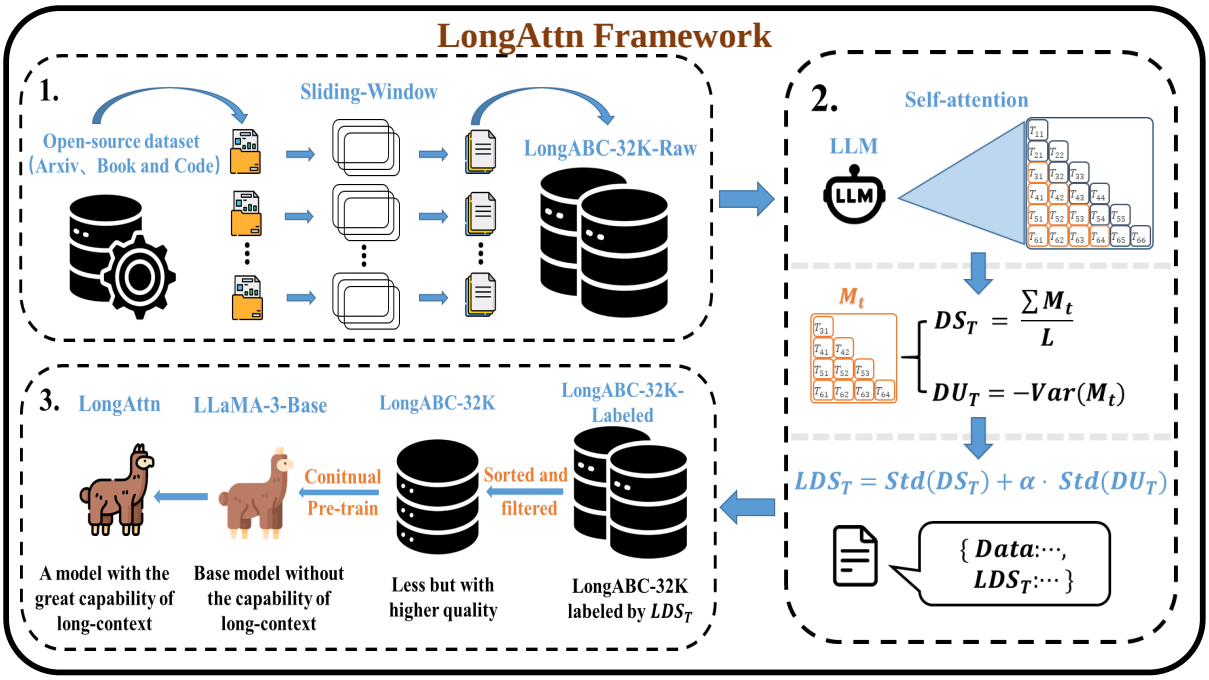


Figure 2: LongAttn Framework: After preprocessing the data, the long-distance dependency strength at the token-level is analyzed using the self-attention mechanism of an LLM. This analysis serves as the basis for filtering the data, which is then used for continual pre-training of a base model that initially lacks long-context capabilities, resulting in our LongAttn model

3.2.1 Token-level Dependency Strength

Given a data instance $s \in \mathcal{D}$, we input it into an LLM and extract the masked self-attention matrix M from the first transformer decoder layer to quantify the long-range dependencies within the data. The choice of using the first layer is driven by two primary reasons: **(1)** It is computationally efficient, requiring approximately 1/32 of the inference time; **(2)** Due to the Attention Sink phenomenon (Xiao et al., 2023), deeper layers of the model tend to disproportionately focus on the initial tokens, irrespective of their semantic relevance to the language modeling task. Consequently, leveraging the shallow layers of the model's decoder is more optimal for capturing the contextual dependencies among tokens in the data.

Define $A_{m,n}$ as the cumulative attention score assigned by n to the first m tokens (i.e., tokens from position 1 to m):

$$A_{m,n} = \sum_{i=1}^{m} M_{i,n} \tag{1}$$

where $M_{i,n}$ represents the attention score assigned by the n-th token to the i-th token. Since the self-attention matrix M has been normalized by the softmax function, it follows that $A_{n,n} = 1$. For the n-th token in the data, where $n > k$, $A_{n-k,n}$ represents the sum of attention scores of all tokens located at least k positions ahead of it. We define the contextual dependency strength of the n-th token as:

$$DS_T^n = \frac{A_{n-k,n}}{A_{n,n}} = A_{n-k,n} \tag{2}$$

which quantifies the proportion of attention scores assigned to tokens at least k positions prior to the n-th token, relative to the total attention scores. For cases where $n \leq k$, we define $DS_T^n = 0$ to account for insufficient context. Finally, the token-level contextual dependency strength of the entire data instance is defined as the average of DS_T^n over all tokens:

$$DS_T = \frac{1}{L} \sum_{i=1}^{L} DS_T^i \tag{3}$$

$$= \frac{1}{L} \sum_{i=k+1}^{L} DS_T^i \tag{4}$$

$$= \frac{1}{L} \sum M_t \tag{5}$$

where L is the total number of tokens in the data and M_t represents the lower triangular matrix in

the bottom left corner of matrix M:

$$M_t = \begin{pmatrix} M_{k+1,1} & 0 & \cdots & 0 \\ M_{k+2,1} & M_{k+2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ M_{L,1} & M_{L,2} & \cdots & M_{L,L-k} \end{pmatrix} \quad (6)$$

3.2.2 Distribution Uniformity of Token Scores

While DS_T provides a measure of dependency strength, it is important to ensure that individual tokens with high scores do not disproportionately influence the overall dependency assessment. For example, In the previously mentioned Attention Sink phenomenon, the first token's scores very high in deeper decoder self-attention layers, which can have a significantly negative impact. Instead, the scores across the entire data segment should be consistently high. To achieve this, we introduce the distribution uniformity of token scores DU_T to measure the uniformity of the score distribution:

$$DU_T = -Variance(M_t) \quad (7)$$

This correction term helps to prevent individual tokens with excessively high attention scores from skewing the overall dependency assessment.

3.2.3 Collaborative Ensemble

To obtain a comprehensive measure of long-range dependencies, we combine the dependency strength DS_T and the distribution uniformity DU_T. Due to the differences in the magnitudes of DU_T and DS_T, as shown as Appendix E, we compute DU_T and DS_T for all data and then standardize them to independent normal distributions. We then use the following formula to calculate the final long-distance dependency score:

$$LDS_T = Std(DS_T) + \alpha \cdot Std(DU_T) \quad (8)$$

where α is a correction factor that balances the contributions of DS_T and DU_T, and Std represents Z-Score standardization.

4 Experimental Setup

4.1 LongAttn Setup and Training Details

In the process of filtering data using LongAttn, we utilize the first transformer decoder layer of LLaMA-3.1 to calculate long-distance dependency score. The length of each data segment L is 32768 and the minimum token distance k is set to $L/4$ (i.e., 8192). We set the correction factor α in the Eq.8 to 0.5.

We adopt Adjusted Base Frequency (ABF) (Xiong et al., 2023) to continual pretrain LLaMA-3, extending the context window size to 32,768 by adjusting the RoPE theta parameter. The continued pre-training is based on the Megatron training framework (Shoeybi et al., 2019), utilizing 8x8 H800 GPUs. Detailed parameters can be found in the Appendix B.1.

4.2 Continual Pre-trained Datasets

We form the following datasets by combining short-context data with selections made through random sampling, the ProLong framework, LongAttn based on LLaMA-3.1-8B, and LongAttn based on LLaMA-3.1-70B: $\mathcal{D}_{Rx}(x \in \{1,3,5,10,20\})$, $\mathcal{D}_{Px}(x \in \{1,3,5,10\})$, $\mathcal{D}_{Ax,8B}(x \in \{1,3,5,10\})$, and $\mathcal{D}_{Ax,70B}(x \in \{1,3,5,10\})$, with x representing the data size in Billions.

To ensure the diversity of the filtered data, we apply the filtering process within each category of datasets separately. For detailed data composition, please refer to the Appendix C.

4.3 Baselines

Data-Scale Comparison To demonstrate the effectiveness of LongAttn, We conduct a data-scale comparison of the long-context retrieval capabilities of models continued pre-trained on $\mathcal{D}_{Rx}(x \in \{1,3,5,10,20\})$, $\mathcal{D}_{Px}(x \in \{1,3,5,10\})$, $\mathcal{D}_{Ax,8B}(x \in \{1,3,5,10\})$, and $\mathcal{D}_{Ax,70B}(x \in \{1,3,5,10\})$.

Fixed-Scale Method Comparison To demonstrate the superiority of LongAttn, we conduct fixed-data method comparison of the models trained on $\mathcal{D}_{Rx}(x \in \{5,10,20\})$, $\mathcal{D}_{P5}$, $\mathcal{D}_{A5,8B}$, and $\mathcal{D}_{A5,70B}$. Additionally, we compare them with similarly sized models that have excellent long-context capabilities. Details of the baselines can be found in Appendix F.

4.4 Evaluation Tasks

We assess the capability of the base model, continually pre-trained within the current window length, based on the following long-context and short-context criteria: **(1)** The best reflection of the base model's long-context capabilities is its long-context retrieval ability, followed by its performance on other long-context tasks. **(2)** No degradation in short-context performance. The evaluation tasks can be divided into the following parts:

Method	Tokens	Niah-Single			Niah-Multikey			Niah-Mult-Value	Niah-Mult-Query	Avg. Score
		Sigle-1	Sigle-2	Sigle-3	MK-1	MK-2	MK-3			
Random	1 B	99.8	**100.0**	93.4	**91.0**	11.6	11.4	**91.7**	93.2	74.0
ProLong		99.4	99.8	92.4	89.2	10.8	24.0	91.6	**93.6**	75.1
LongAttn-8		**100.0**	**100.0**	91.4	88.6	16.2	19.2	90.3	93.4	74.9
LongAttn-70		**100.0**	**100.0**	**95.4**	88.0	**29.0**	**35.0**	90.4	92.4	**78.8**
Random	3 B	**100.0**	**100.0**	86.2	92.8	62	8.6	70.0	95.9	76.9
ProLong		**100.0**	99.8	79.6	**93.8**	**60.4**	**32.0**	85.9	95.0	80.8
LongAttn-8		**100.0**	**100.0**	88.8	92.2	60.0	31.2	79.7	94.7	80.8
LongAttn-70		**100.0**	**100.0**	**91.6**	93.6	57.4	19.8	**88.8**	**96.2**	**80.9**
Random	5 B	**100.0**	99.8	81.8	**94.8**	56.4	11.8	84.4	96.5	78.2
ProLong		**100.0**	**100.0**	78.0	92.8	64.8	40.4	77.8	95.9	81.2
LongAttn-8		**100.0**	99.8	81.6	92.4	62.6	37.2	**87.6**	**97.3**	82.3
LongAttn-70		**100.0**	**100.0**	**83.8**	92.8	**84.8**	**46.8**	78.8	95.2	**85.2**
Random	10 B	**100.0**	**100.0**	84.0	92.6	58.2	14.2	90.9	**96.9**	79.6
ProLong		**100.0**	**100.0**	83.4	92.8	74.4	32.2	88.7	95.5	83.4
LongAttn-8		**100.0**	**100.0**	**87.4**	**93.0**	72.2	23.0	**93.1**	96.8	83.2
LongAttn-70		**100.0**	**100.0**	86.8	92.4	**80.6**	**34.4**	92.0	96.5	**85.3**
Random	20 B	100.0	100.0	84.6	91.0	66.2	22.4	93.3	96.5	81.8

Table 1: Models trained with different methods for selecting varying scales of tokens were evaluated on complex NIAH tasks. Random, ProLong, LongAttn-8, and LongAttn-70 represent random selection, selection based on the ProLong framework, selection based on LongAttn with LLaMA-3.1-8B, and selection based on LongAttn with LLaMA-3.1-70B, respectively. And **bold** number is used to highlight the better-performing models within each data size category.

Model	Trained Dataset	Short-Context Task				Avg.
		MMLU	HS	HE	OBQA	
LLaMA -3-Base	†	**65.9**	49.9	25.0	72.0	53.2
	$\mathcal{D}_{R5}$	61.8	52.4	19.5	81.8	53.9
	$\mathcal{D}_{P5}$	61.0	38.3	23.2	79.4	50.5
	$\mathcal{D}_{A5,8B}$	61.6	47.1	25.6	**82.6**	54.2
	$\mathcal{D}_{A5,70B}$	61.0	**52.8**	**28.1**	80.4	**55.6**

Table 2: The short-context fundamental capabilities of our continued pre-trained models and LLaMA-3-base. † indicates no training. MMLU, HS, HE, and OBQA stand for the MMLU, HellaSwag, HumanEval, and OpenBookQA tasks, respectively.

Long-context Retrieval Retrieval ability is the most crucial and best reflects the model's long-context ability before post-training. The 'Needle In A Haystack' task analysis in-context retrieval ability of long-context LLMs. The original 'needle in a haystack' task was relatively simple. RULER (Hsieh et al., 2024) introduced a more detailed and complex 'needle in a haystack' task, and we use RULER with a length of 32k to comprehensively evaluate long-context retrieval ability.

Long-context Benchmark In addition to retrieval ability, we also want to evaluate the model's performance on formal long-context tasks. LongBench (Bai et al., 2023b) is the first proposed bilingual long-context benchmark, which includes a total of 21 tasks categorized into 6 main types, with task lengths ranging from about 0 to 20k. RULER provides longer, variable-length evaluations across 13 complex tasks. Here, we will evaluate the tasks at the 32k length to assess changes in the model's long-context capabilities.

Fundamental Abilities of LLMs. We use HumanEval (Chen et al., 2021) to assess code evaluation capability and OpenBookQA (Mihaylov et al., 2018) to assess book knowledge extraction ability. Additionally, we use Hellaswag(Zellers et al., 2019) and MMLU (Hendrycks et al., 2020) to assess its broader short-context fundamental capabilities.

5 Experimental Results

We validate the **effectiveness**, **scalability**, and **high efficiency** of LongAttn through a series of data-scale and Lateral comprehensive experiments.

5.1 Performance on Retrieval Ability

We evaluate the retrieval capabilities of models trained with LongAttn-selected data and compare them with models trained on randomly selected data and ProLong-selected data. The results are shown in Table 1. The models trained with

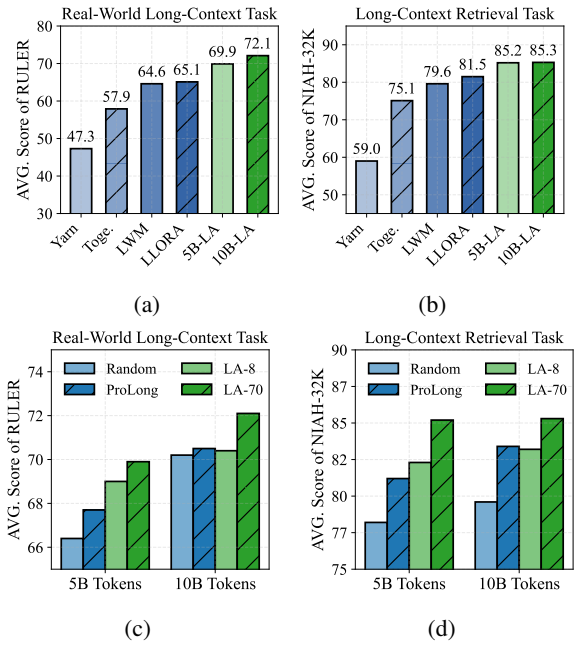


Figure 3: **(a)** and **(b)** show the performance of other long-context LLMs and LongAttn-trained models on the RULER and complex NIAH tasks. **(c)** and **(d)** show the performance of models trained with different methods on the same tasks. Toge. and LLORA represent Together and LongLORA, respectively. 5B-LA and 10B-LA represent models trained on 5B and 10B tokens selected by LongAttn. LA-8 and LA-70 represent LongAttn based on 8B and 70B models, respectively.

LongAttn-selected data consistently outperform those trained on randomly selected or ProLong-selected data across all data scales, demonstrating the effectiveness of LongAttn in improving data quality for long-context modeling.

Notably, models trained on a smaller amount of data filtered using our method even outperform those trained on a larger amount of randomly selected data in retrieval tasks. For example, the model trained on just 5B tokens filtered by LongAttn outperforms models trained on 10B or even 20B randomly selected tokens. This indicates that LongAttn can significantly enhance the efficiency of data usage for long-context pre-training.

5.2 Performance on Fundamental Abilities

The results in Table 2 indicate that data selected by LongAttn not only maintains the model's short-context capabilities but enhances them in specific domains. For example, the LongABC-32K-Raw dataset includes book and code data, and our model performs well on short-context tasks such as OpenBookQA (Mihaylov et al., 2018) and HumanEval (Chen et al., 2021).

However, there is a slight decline in performance on MMLU (Hendrycks et al., 2020). This is expected, as we do not include such data during continual pre-training, so the base model experienced some forgetting in these areas.

5.3 Performance on Long-context Benchmark

As shown in Figure 3a and 3c, models trained on data filtered by LongAttn outperform those trained on equivalent amounts of data selected randomly or by ProLong. LongAttn's performance is also comparable to models trained on larger data volumes. Additionally, on the RULER-32K benchmark, LongAttn outperforms all other long-context models of similar parameter sizes.

As shown in Table 3, we compare model performance on LongBench, which consists of 21 evaluation tasks. We calculate the average score for each of the six categories to represent overall performance. The results show that LongAttn outperforms models trained on equivalent data selected randomly or by ProLong in almost all tasks and even surpasses models trained on larger amounts of randomly selected data. However, while 5B data selected by LongAttn-70 outperforms 10B randomly selected data, it does not perform as well as 5B data selected by LongAttn-8. We speculate this is because the average context length in LongBench is far below 32k, thus not effectively showcasing the advantage of 5B data selected by LongAttn-70.

6 Analysis

6.1 Ablation Study

To investigate the impact of the constraint factor α and the correction term DU_T on regulating LDS_T, we conduct ablation experiments on the $\mathcal{D}_{A3}$ and $\mathcal{D}_{A5}$ datasets using retrieval tasks. The default setting of the constraint factor α is 0.5.

As shown in the table 4, we can see that the correction term DU_T plays a positive role in the data selection results. In addition, the constraint on the dependency strength DS_T by DU_T should not be too large, which suggests that the constraint on DS_T by DU_T should be moderate to avoid over-correction.

6.2 The Scalability of LongAttn

Figures 3c and 3d show that LongAttn significantly improves performance when using stronger models. This indicates that more powerful models can better analyze the dependencies between long-context

Method	Number of Tokens Trained	Single-Doc QA	Multi-Doc QA	Summri-zation	Few-shot Learning	Synthetic Tasks	Code Com-pletion	Avg. Score
Trained on 5B Tokens from Different Methods								
Random	5B	10.11	6.57	13.72	64.10	1.83	65.05	24.46
ProLong	5B	11.95	**12.59**	17.87	63.33	4.15	65.01	26.93
LongAttn-8	5B	**13.01**	11.20	18.96	**64.62**	**5.12**	**65.06**	**27.46**
LongAttn-70	5B	12.39	9.33	**19.72**	64.1	3.42	65.03	26.78
Trained on over 5B Tokens Selected Randomly								
Random	10B	9.41	8.93	19.30	63.89	4.83	65.57	26.27
Random	20B	11.45	11.72	20.41	64.13	9.67	66.51	28.23

Table 3: The performance of models continued pre-trained using data filtered by different methods on LongBench. Random, ProLong, LongAttn-8, and LongAttn-70 represent data selected randomly, data selected using the ProLong framework, data selected by the LongAttn framework with LLaMA-3.1-8B, and data selected by the LongAttn framework with LLaMA-3.1-70B, respectively.

Model	RULER-NIAH-32K
$LongAttn_{\mathcal{D}_{A3}}$	**80.83**
w/ $\alpha = 1$	79.49(-1.34)
w/o DU_T	78.28(-2.55)
$LongAttn_{\mathcal{D}_{A5}}$	**82.30**
w/ $\alpha = 1$	81.05(-1.25)
w/o DU_T	82.11(-0.19)

Table 4: Ablation experiments on the constraint factor α and the correction term DU_T were conducted on the RULER-NIAH-32K task.

Method	Model	GPU Hours
ProLong	OPT-350M	30
	LLaMA-3.1-8B	600
LongAttn	LLaMA-3.1-8b	50
	LLaMA-3.1-70b	100

Table 5: Compared the GPU hours used by different methods on LongABC-32K-Raw, using H800 GPUs. For implementation simplicity, we used the traditional attention computation method in LongAttn. If efficient methods like Flash-attn were adopted, the speed would further improve.

tokens. It can be envisioned that using LongAttn with larger models could yield even stronger performance.

However, in works like ProLong, computational efficiency is constrained by the approach, making it unfeasible to use larger models. This unique advantage of LongAttn highlights its tremendous growth potential.

6.3 The Efficiency of LongAttn

Compared to sentence-level methods like ProLong, LongAttn is significantly more efficient. ProLong divides the data into sentence segments and calculates the relative perplexity and distance between each segment, which is computationally expensive, especially for LLMs. As a result, only smaller models are used in their work. In contrast, LongAttn only requires a single inference pass to obtain relative scores between all tokens, using just the first layer of the LLM's decoder. This approach is far more efficient and scalable.

Table 5 compares the GPU hours consumed by the two methods using models of different sizes on the LongABC-32K-Raw dataset. LongAttn, even with the traditional attention computation method, is much faster than ProLong. If more efficient methods like Flash-attention were adopted, the speed of LongAttn could be further improved.

7 Conclusion

In this paper, we introduce LongAttn, a framework evaluates long-range dependencies at the token level. LongAttn is effective as the self-attention mechanism captures relationships between all token contexts during inference. This approach to measuring long-range dependencies aligns better with the underlying operating principles of LLMs. We validate the effectiveness, scalability, and high efficiency of LongAttn through a series of comprehensive experiments. Additionally, our research contributes to the previously limited study of high-quality long-context training data. This finding suggests promising directions for future research, and we anticipate further advancements in this domain through subsequent investigations.

Limitations

Although LongAttn has demonstrated satisfactory performance, there is still room for improvement. Specifically, we used the traditional attention map calculation method, which is inefficient. While its efficiency is satisfactory, there is still significant potential for enhancement. In future work, we hope to overcome the shortcomings, refine our method further, and advance the development of long-context capabilities in LLMs.

Ethics Statement

This work fully complies with the ACL Ethics Policy. We declare that there are no ethical issues in this paper, to the best of our knowledge.

References

Chenxin An, Jun Zhang, Ming Zhong, Lei Li, Shansan Gong, Yao Luo, Jingjing Xu, and Lingpeng Kong. 2024. Why does the effective context length of llms fall short? *arXiv preprint arXiv:2410.18745*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023a. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. 2024. LongAlign: A recipe for long context alignment of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1376–1395, Miami, Florida, USA. Association for Computational Linguistics.

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

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Longze Chen, Ziqiang Liu, Wanwei He, Yunshui Li, Run Luo, and Min Yang. 2024a. Long context is not long at all: A prospector of long-dependency data for large language models. *arXiv preprint arXiv:2405.17915*.

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. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

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

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

Zhi Chen, Qiguang Chen, Libo Qin, Qipeng Guo, Haijun Lv, Yicheng Zou, Wanxiang Che, Hang Yan, Kai Chen, and Dahua Lin. 2024b. What are the essential factors in crafting effective long context multi-hop instruction datasets? insights and best practices. *ArXiv*, abs/2409.01893.

Colin B. Clement, Matthew Bierbaum, Kevin P. O'Keeffe, and Alexander A. Alemi. 2019. On the use of arxiv as a dataset. *Preprint*, arXiv:1905.00075.

Harm de Vries. 2023. In the long (context) run. https://www.harmdevries.com/post/context-length/.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. Longrope: Extending llm context window beyond 2 million tokens. In *Forty-first International Conference on Machine Learning*.

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

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling. *Preprint*, arXiv:2101.00027.

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

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

Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. 2024. Llm maybe longlm: Selfextend llm context window without tuning. In *Forty-first International Conference on Machine Learning*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Huan Yee Koh, Jiaxin Ju, Ming Liu, and Shirui Pan. 2022. An empirical survey on long document summarization: Datasets, models, and metrics. *ACM computing surveys*, 55(8):1–35.

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

Kai Lv, Xiaoran Liu, Qipeng Guo, Hang Yan, Conghui He, Xipeng Qiu, and Dahua Lin. 2024. Longwanjuan: Towards systematic measurement for long text quality. *arXiv preprint arXiv:2402.13583*.

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

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. 2023. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only. *Preprint*, arXiv:2306.01116.

Bowen Peng and Jeffrey Quesnelle. 2023. Ntk-aware scaled rope allows llama models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation. https://www.reddit.com/r/LocalLLaMA/comments/14lz7j5/ntkaware_scaled_rope_allows_llama_models_to_have.

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

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, et al. 2024. Dolma: An open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*.

Together.Ai. 2023. Preparing for the era of 32k context: Early learnings and explorations, 2023a. https://www.together.ai/blog/llama-2-7b-32k.

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

Liang Wang, Nan Yang, Xingxing Zhang, Xiaolong Huang, and Furu Wei. 2024a. Bootstrap your own context length. *arXiv preprint arXiv:2412.18860*.

Minzheng Wang, Longze Chen, Fu Cheng, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, et al. 2024b. Leave no document behind: Benchmarking long-context llms with extended multi-doc qa. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5627–5646.

Maurice Weber, Daniel Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, et al. 2024. Redpajama: an open dataset for training large language models. *arXiv preprint arXiv:2411.12372*.

Wenhao Wu, Yizhong Wang, Yao Fu, Xiang Yue, Dawei Zhu, and Sujian Li. 2024. Long context alignment with short instructions and synthesized positions. *arXiv preprint arXiv:2405.03939*.

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

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

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

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

A Algorithm for Pre-process

Algorithm 1 Sliding Window Sample Algorithm

function SLIDINGWINDOW($data, W$)
  if len($data$) $< W$ **then**
    return $\emptyset$
  end if
  $l \leftarrow 0$
  $r \leftarrow$ len($data$)
  $S \leftarrow \emptyset$
  while $r - l > 3W$ **do**
    $S \leftarrow S \cup \{data[l : l + W]\}$
    $l \leftarrow l + W$
    $S \leftarrow S \cup \{data[r - W : r]\}$
    $r \leftarrow r - W$
  end while
  $\Delta \leftarrow r - l$
  if $W < \Delta \leq 2W$ **then**
    $S \leftarrow S \cup \{data[l : l + W], data[r - W : r]\}$
  else if $2W < \Delta \leq 3W$ **then**
    $m \leftarrow l + \lfloor(\Delta - W)/2\rfloor$
    $S \leftarrow S \cup \{data[l : l + W], data[m : m + W], data[r - W : r]\}$
  end if
  return S
end function

The Algorithm 1 demonstrates how we perform sliding window pre-processing on the data. The length of the data processed using this method will remain consistent with the window size, and compared to the truncation method, this algorithm better preserves the completeness of the original information.

B Training Details

B.1 Training Parameters

The specific experimental parameters for continual pre-training using Megatron (Shoeybi et al., 2019) are shown in Table 6.

Params	Methods		
	Random	**ProLong**	**LongAttn**
learning rate(lr)	1×10^{-5}	1×10^{-5}	1×10^{-5}
lr decay style	cosine	cosine	cosine
GPUs (H800)	64	64	64
mbs	1	1	1
gas	8	8	8
tp size	8	8	8
pp size	1	1	1
dropout	0.1	0.1	0.1
seq length	32768	32768	32768

Table 6: Parameter settings for continual pre-training by different methods based on the Megatron framework.

B.2 Training Dataset

When continuing pre-training, we use the data ratios shown as Table 7, where ArXiv, Book, and Code data refer to the data selected through different methods (random selecting, based on the ProLong (Chen et al., 2024a) framework, or based on the LongAttn framework).

Types	**length**	**Source**	**Ratio**
Wiki	Short	Dolma (Soldaini et al., 2024)	3%
Github	Short	Pile (Gao et al., 2020)	3%
Web	Short	Refinedweb (Penedo et al., 2023)	4%
ArXiv	Long	LongABC-Arxiv	30%
Book	Long	LongABC-Book	30%
Code	Long	LongABC-Code	30%

Table 7: The types of data and their proportions used during the continuation of pre-training. LongABC-Arxiv, LongABC-Book, and LongABC-Code refer to the types of data selected using different methods from LongABC-32K-Raw.

C Details of Continual Pre-train Dataset

As shown as Figure 8, LongABC-32K-Raw is a dataset obtained by quantitatively sampling long text data and then preprocessing it as mentioned in 3.1.

LongABC-32K-Raw serves as the data source. We filter it using different methods, including random selecting, selecting based on the ProLong framework, and selecting based on the LongAttn framework. The filtered data is then combined with quantified short-context data to form our pre-training dataset, as shown in Table 7.

Category	**Source**	**Scale**
ArXiv	ArXiv (Clement et al., 2019)	12B Tokens
Book	Dolma (Soldaini et al., 2024), RedPajama (Weber et al., 2024)	12B Tokens
Code	Dolma (Soldaini et al., 2024)	12B Tokens

Table 8: Data source of LongABC-32K-Raw and composition of its various parts.

D Other Experimental Results

The evaluation results on RULER for models trained with data selected from LongABC-32K-

Method	Tokens	Retrival	VT	Aggregation			QA			Avg.
		Avg.		CWE	FWE	Avg.	QA1	QA2	Avg.	Score
Trained on 5B Tokens from Different Methods										
Random	5B	78.2	40.6	31.4	66.7	**49.0**	55.2	43.8	49.5	66.4
ProLong		81.2	**51.8**	13.0	65.4	39.2	57.2	43.4	**50.3**	67.7
LongAttn-8		82.3	50.3	19.8	71.0	45.4	53.4	44.0	48.7	69.0
LongAttn-70		**85.2**	43.4	16.8	68.5	42.7	55.6	43.0	49.3	**69.9**
Trained on 10B Tokens from Different Methods										
Random	10B	79.6	48.8	53.3	74.2	**63.7**	55.4	43.6	49.5	70.2
ProLong		83.4	55.1	19.4	76.8	48.1	54.6	44.6	49.6	70.6
LongAttn-8		83.2	52.1	21.8	77.9	49.9	54.6	43.8	49.2	70.4
LongAttn-70		**85.3**	**55.6**	31.9	67.4	49.7	55.4	44.0	**49.7**	**72.1**
Trained on 20B Tokens Selected Randomly										
Random	20B	81.8	47.4	51.9	87.9	69.9	51.9	56.0	46.4	73.0

Table 9: The performance of models continued pre-trained using data filtered by different methods on RULER. Random, ProLong, LongAttn-8, and LongAttn-70 represent data selected randomly, data selected using the ProLong framework, data selected by the LongAttn framework with LLaMA-3.1-8B, and data selected by the LongAttn framework with LLaMA-3.1-70B, respectively.

Raw using different methods are shown in Table 9. RULER includes 13 tasks, categorized into four major types: retrieval ability, multi-hop tracking ability, information aggregation ability, and question answering ability. The retrieval ability has been thoroughly evaluated earlier, so only the average score is presented here.

E Distribution of DS_T and DU_T

Statistical Indicators	Arxiv		Book		Code	
	DS_T	DU_T	DS_T	DU_T	DS_T	DU_T
Min Val.	0.25	2.2×10^{-7}	0.21	1.6×10^{-7}	0.18	9.7×10^{-8}
Max Val.	0.50	1.8×10^{-6}	0.59	4.9×10^{-6}	0.54	2.4×10^{-6}
Mean	0.43	8.5×10^{-7}	0.40	4.8×10^{-7}	0.39	6.1×10^{-7}

Table 10: Statistical indicators of DS_T and DU_T after evaluating LongABC-32K-Raw using the LongAttn framework based on LLaMA-3.1-70B

The distribution DS_T and DU_T measured by LongAttn based on LLaMA-3.1-70B is shown in Table 10. They are distributed across different value ranges.

F Baselines

The specific models and baselines for our data-scale and fixed-data method comparison experiments are detailed in Table 11.

Comparison Method	Base Model	Trained Dataset	Selected Method	Tokens
Data-Scale	LLaMA-3	$\mathcal{D}_{Rx}$	Selected Randomly	$x \in \{1B, 3B, 5B$,10B,20B$\}$
		$\mathcal{D}_{Px}$	ProLong	$x \in \{1B, 3B, 5B, 10B\}$
		$\mathcal{D}_{Ax,8B}$	LongAttn-8	$x \in \{1B, 3B, 5B, 10B\}$
		$\mathcal{D}_{Ax,70B}$	LongAttn-70	$x \in \{1B, 3B, 5B, 10B\}$
Fixed-Scale Method	LLaMA-3	$\mathcal{D}_{Rx}$	Selected Randomly	$x \in \{5B, 10B, 20B\}$
		$\mathcal{D}_{Px}$	ProLong	$x \in \{5B, 10B\}$
		$\mathcal{D}_{Ax,8B}$	LongAttn-8	$x \in \{5B, 10B\}$
		$\mathcal{D}_{Ax,70B}$	LongAttn-70	$x \in \{5B, 10B\}$
	Yarn (Peng et al., 2023)	†	†	†
	LWM (Liu et al., 2024)	†	†	†
	Together (Together.Ai, 2023)	†	†	†
	LongLORA (Chen et al., 2023b)	†	†	†

Table 11: The experiments compared different models and baselines. **Selected Method** indicates the method used to filter the current training set, and **Tokens** represents the number of tokens used for training. † indicates the absence of a given option. ProLong, LongAttn-8, and LongAttn-70 represent the ProLong framework, LongAttn based on LLaMA-3.1-8B, and LongAttn based on LLaMA-3.1-70B, respectively.