NExtLong: Toward Effective Long-Context Training without Long Documents

Chaochen Gao[1,2], **Xing Wu**[1,2,3] ✉, **Zijia Lin**[4], **Debing Zhang**[3], **Songlin Hu**[1,2] ✉
[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
[3]Xiaohongshu Inc, [4]Tsinghua University
{gaochaochen,wuxing,husonglin}@iie.ac.cn
linzijia07@tsinghua.org.cn, dengyang@xiaohongshu.com

Abstract

Large language models (LLMs) with extended context windows have made significant strides yet remain a challenge due to the scarcity of long documents. Existing methods tend to synthesize long-context data [1] but lack a clear mechanism to reinforce the long-range dependency modeling. To address this limitation, we propose NExtLong, a novel framework for synthesizing long-context data through Negative document Extension. NExtLong decomposes a document into multiple meta-chunks and extends the context by interleaving hard negative distractors retrieved from pretraining corpora. This approach compels the model to discriminate long-range dependent context from distracting content, enhancing its ability to model long-range dependencies. Extensive experiments demonstrate that NExtLong achieves significant performance improvements on the HELMET and RULER benchmarks compared to existing long-context synthesis approaches and leading models, which are trained on non-synthetic long documents. These findings highlight NExtLong's ability to reduce reliance on non-synthetic long documents, making it an effective framework for developing advanced long-context LLMs. Our code is available in https://github.com/caskcsg/longcontext/tree/main/NExtLong.

1 Introduction

Large language models (LLMs) have garnered significant attention due to their powerful and versatile capabilities. Recently, the context length of LLMs has been rapidly extended (Peng et al., 2023; AI et al., 2024; Yang et al., 2024). For example, the Llama series models increase the context length from 4k in Llama 2 (Touvron et al., 2023b) to 128K in Llama 3.1 (Meta, 2024). The increased context window enables LLM to unlock more challenging tasks, such as Document Summary (Wu et al., 2023b), Longbook QA (Caciularu et al., 2023) and Code Planning (Bairi et al., 2023). To model long-range dependencies, mainstream methods (Fu et al., 2024; Gao et al., 2024b) typically continue training existing LLMs pre-trained on a 4K or 8K context length with long documents that reach the target length, e.g., 128K. However, the scarcity of high-quality long documents in most domains remains a significant challenge, particularly as the target context length continues to increase (Gao et al., 2024a).

To address the challenge of scarcity of long documents, existing approaches synthesize long-context data by concatenating shorter documents. Similarity-based methods, such as KNN (Guu et al., 2020; Levine et al., 2021), aggregate the top-k se-

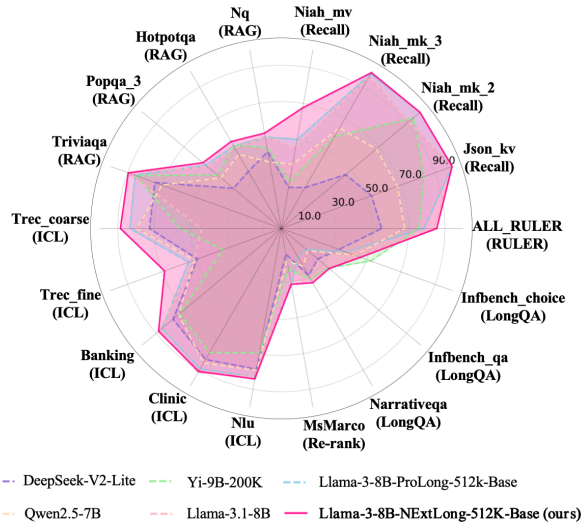

Figure 1: Comparison of existing remarkable models and NExtLong on the HELMET(Yen et al., 2024b) and RULER(Hsieh et al., 2024) benchmarks. We evaluate various task types classified by HELMET. All results are averaged over sequence lengths of 8K, 16K, 32K, 64K, and 128K.

[1]In this work, we define "long-context data" as synthetic long-form datasets, and "long documents" as non-synthetic long documents that meet the target training length.

mantically similar short documents into a longer document. Other studies (Roziere et al., 2023; Ouyang et al., 2022; Touvron et al., 2023a) randomly sample and concatenate short documents, often compromising coherence and relevance. Recently, Quest (Gao et al., 2024a) aims to balance semantic correlation and contextual diversity by retrieving documents relevant to specific keywords. However, those methods typically concatenate short documents based on random or similarity-based rankings, lacking a clear mechanism for capturing long-range dependencies.

An intuitive approach to building documents with long-range dependencies is to insert additional text between dependent segments (Tian et al., 2024), thereby transforming short dependencies into long-range ones. However, recent studies show that large language models can be easily distracted by irrelevant context (Shi et al., 2023a), and this issue is exacerbated as the context length increases(Han et al., 2023). This raises a critical challenge: *how can we enhance a model's ability to discriminate long-range dependent information from distracting content within extended contexts?*

Inspired by the hard negative technique (Robinson et al., 2020; Kalantidis et al., 2020; Zhan et al., 2021) from contrastive learning, which introduces hard negatives to enhance a model's ability to discriminate relevant samples from distracting ones, we adapt this concept to create hard negative distractors that reinforce long-range dependency modeling. The key idea is to generate negative-extended documents by inserting semantically similar yet distracting texts between dependent fragments. These distractions increase the model's learning difficulty, thereby enhancing its capacity to model long-range dependencies. Specifically, NExtLong works by first chunking a document into multiple chunks, termed meta-chunks. We retrieve hard negatives from a pretraining corpus for meta-chunks and interleave them between dependent meta-chunks. Since the pretraining corpus undergoes extensive deduplication, these hard negatives share partial semantic similarities with the meta-chunks but do not replicate their content. By inserting these distractors between originally dependent meta-chunks, NExtLong not only increases the distance between dependent chunks—effectively transforming the dependencies into long-range ones—but also introduces distracting noise, which compels the model to reinforce its ability to discriminate long-range dependent context from distracting content.

Extensive experiments on the HELMET(Yen et al., 2024b) and RULER(Hsieh et al., 2024) benchmarks demonstrate that NExtLong significantly improves the model's ability to capture and utilize long-range dependencies. Overall, NExtLong delivers an average performance improvement of 7.33% over the previous long-context synthesis method Quest (Gao et al., 2024a). Moreover, compared to existing remarkable models trained by long documents, NExtLong achieves extraordinary results, as highlighted in Figure 1. These results demonstrate that NExtLong is a highly effective method for synthesizing long-context data. The synthesized data significantly alleviates the dependence on training large long-context models on long documents and holds the potential to train ultra-long context models that are not constrained by the scarcity of long documents.

Our main contributions can be summarized as follows:

- We propose NExtLong, a simple and effective method that extends the document to strengthen the model's ability to model long-range dependencies.
- We provide an in-depth analysis of the key components that contribute to the effectiveness of NExtLong.
- Our experiments show that NExtLong achieves a significant improvement across multiple long-context evaluation tasks, demonstrating the effectiveness of negative document extension in training long-context LLMs.

2 Related Work

Unlocking LLMs' ability to process long-context tasks. **Train-free methods** bypass parameter updates for long-context handling. LM-Infinite (Han et al., 2023) employs a Λ-shaped attention mask with a distance limit for length generalization. StreamingLLM (Xiao et al., 2023) mitigates the "attention sink" phenomenon by balancing attention scores. Self-Extend (Jin et al., 2024b) introduces group-wise attention to map unseen relative positions, while DCA (An et al., 2024) uses token-wise attention and memory-efficient mechanisms for effective context extension. **Train-based methods** enhance performance through continued training. Chen et al. (2023b) extend RoPE-based (Su et al., 2021) LLMs via positional interpolation, and PoSE

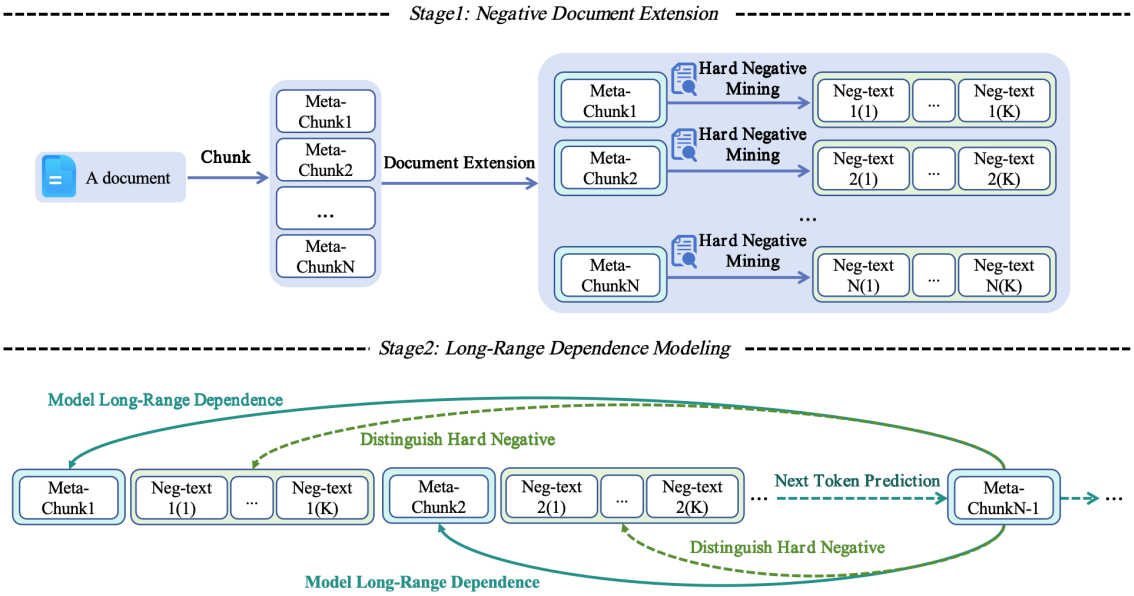

Figure 2: The NExtLong method consists of two stages. In the first stage, a document is chunked into multiple meta-chunks, and each meta-chunk is mined for numerous hard negatives. These hard negatives are then concatenated with the meta-chunks to create a long document. In the second stage, the model is trained using this synthesized long document, focusing on modeling long-range dependencies by identifying the meta-chunks across a wide range of hard negatives.

(Zhu et al., 2023) applies positional skip-wise training to decouple training and target lengths. Recently, upsampling long documents across diverse domains has emerged as a critical factor in advancing long-context modeling (Fu et al., 2024; Gao et al., 2024b; Xiong et al., 2023).

However, those train-based methods are dependent on the availability of high-quality long documents, which are scarce in many domains and become increasingly harder to obtain as context lengths grow. In contrast, NExtLong overcomes this challenge by extending documents with hard negatives, alleviating the reliance on naturally occurring long documents.

Synthesizing long-context texts by concatenating short documents. Past approaches to synthesizing long-context data primarily rely on concatenating short documents. Those methods often lack a mechanism to ensure that the concatenated documents maintain explicit long-range dependencies. Some methods randomly sample and concatenate short documents (Roziere et al., 2023; Chen et al., 2023c), while others attempt to preserve semantic coherence by clustering similar documents using KNN (Guu et al., 2020). Recent works like Quest (Gao et al., 2024a) try to balance semantic relevance and diversity by retrieving keyword-related documents. However, those methods typically fail to hold a mechanism to effectively model long-range dependencies across distant documents. In contrast, NExtLong not only synthesizes long-context data but also explicitly introduces hard negatives between document chunks, which reinforces the model's ability to learn and utilize long-range dependencies.

Hard negative technique. Hard negative mining is a well-established technique in contrastive learning and dense retrieval, aimed at improving model discrimination by introducing samples that are semantically similar yet incorrect (Robinson et al., 2020; Xiong et al., 2020). In dense retrieval, it enables the model to effectively distinguish between relevant and irrelevant information by selecting challenging negative samples (Zhan et al., 2021; Wu et al., 2023a). While traditionally applied to retrieval tasks, recent studies explore its potential in LLMs, such as Jin et al. (2024a), which investigates the role of hard negatives in Retrieval-Augmented Generation (RAG). However, those approaches have not yet been applied to document synthesis for adapting LLMs to handle longer contexts. In this work, we adapt hard negative mining for document synthesis by interleaving hard negatives between meta-chunks in NExtLong, which helps the model better focus on long-range dependent context and improves its ability to process

long-context tasks.

3 Method

This section introduces our proposed method, NExtLong, which comprises two stages: **Negative Document Extension** and **Long-Range Dependence Modeling**. NExtLong aims to enhance long-context modeling by synthesizing extended-length documents. An overview of the approach is shown in Figure 2, and the corresponding pseudocode is presented in Appendix C.2.

3.1 Negative Document Extension

The Negative Document Extension stage consists of two steps: document chunking and hard negative mining.

3.1.1 Document Chunking

We sample a document from the training dataset as a meta-document r and divide it into sequential meta-chunks. We define the documents to be expanded as meta-documents. The meta-document is divided into several chunks according to a certain chunking granularity. These chunks are defined as meta-chunks. To ensure sentence integrity, we define a maximum length s as the chunking granularity. The chunking process follows a two-step approach:

1. **Splitted by newline:** The meta-document r is first splitted into paragraphs based on newline characters (\n), preserving the coherence of each paragraph.

2. **Form chunks:** These paragraphs are concatenated sequentially to form meta-chunks m_i until the cumulative length reaches the maximum length s. If adding another paragraph exceeds this threshold, the current group is finalized as a complete meta-chunk, and the process continues with the remaining text. The effect of chunking granularity s is analyzed in Section 5.5.

In this way, the meta-document r is divided into p meta-chunks:

$$r \xrightarrow{\text{chunk}} \{m_1, m_2, \ldots, m_p\} \tag{1}$$

The number of meta-chunks p depends on the length of the meta-document r and the chunking granularity s.

3.1.2 Hard Negative Mining

To obtain distracting texts as hard negatives for each meta-chunk, we build a FAISS index from the pretraining dataset, which undergoes extensive deduplication. Unlike methods that treat entire documents as indivisible units, we also divide each document in the pretraining dataset into smaller chunks based on the same granularity s. This chunking enables more fine-grained and efficient content retrieval during the extension process. Formally, each document d_i in the pretraining corpus is divided into q chunks:

$$d_i \xrightarrow{\text{chunk}} \{c_{i_1}, c_{i_2}, \ldots, c_{i_q}\} \tag{2}$$

Each chunk is indexed individually for precise and efficient retrieval. We compute the embedding vector e_i for each chunk and insert it into the FAISS index:

$$\{c_{i_1}, \ldots, c_{i_q}\} \xrightarrow{\text{project}} \{e_{i_1}, \ldots, e_{i_q}\} \xrightarrow{\text{index}} \text{FAISS} \tag{3}$$

After building the FAISS index, we retrieve the top-k most similar chunks as hard negatives n_{i_j} for each meta-chunk m_i. These hard negatives are then concatenated with the meta-chunk to form an extended chunk l_i:

$$l_i = [m_i, n_{i_1}, n_{i_2}, \ldots, n_{i_k}] \tag{4}$$

We conduct ablation experiments on the position of meta-chunks (Appendix A.1), confirming that placing the meta-chunk before the hard negatives yields better performance. The number of hard negatives, i.e., k, depends on the length of the meta-document, chunking granularity s, and target context length. Details for calculating k are provided in Appendix C.1.

Finally, we synthesize a long document t by concatenating the extended chunks:

$$t = [l_1, l_2, \ldots, l_p] \tag{5}$$

3.2 Long-Range Dependence Modeling

In alignment with the pretraining stage, we employ next token prediction (NTP) loss (Radford, 2018) to extend the context length of the base model. The loss function is defined as:

$$\text{Loss} = -\sum_{t=1}^{T} \log P(x_{t+1}|x_1, x_2, \ldots, x_t) \tag{6}$$

The key distinction of NExtLong lies in the differentiation of tokens during training. The tokens

in the synthesized long document t are classified into two categories: meta-chunks m_i and hard negatives n_{i_j}. Together, they form an extended chunk l_i. For simplicity, we use m_i, n_{i_j}, and l_i to denote the encoded tokens of meta-chunks, hard negatives, and extended chunks, respectively. The loss function can thus be reformulated as:

$$\begin{aligned}\text{Loss} &= -\sum_{t=1}^{T} \log P(x_{t+1}|m_1, n_{1_1}, n_{1_2}, \ldots, x_t) \\ &= -\sum_{t=1}^{T} \log P(x_{t+1}|l_1, l_2, \ldots, x_t)\end{aligned} \tag{7}$$

The NTP loss encourages the model to distinguish relevant meta-chunks from surrounding hard negatives and to model long-range dependencies effectively. In Section 4 and Section 5, we empirically demonstrate that incorporating hard negatives in the loss function improves the model's ability to model long-range dependencies across extensive contexts.

4 Experiments

In this section, we evaluate the effectiveness of NExtLong by comparing it with other data synthesis methods (Section 4.2) and state-of-the-art (SOTA) models (Section 4.3).

4.1 Experimental Setups

Datasets We select two commonly used pretraining datasets composed entirely of short documents (Refer to Appendix A.2 for document length distribution): Cosmopedia v2 (Ben Allal et al., 2024) and FineWeb-Edu (Lozhkov et al., 2024). Both datasets are used for the main experiments, and we also provide ablation studies on their selection in Appendix A.3. Various methods, including NExtLong and baseline approaches, are employed to synthesize target-length samples concatenated from these short documents. The datasets are described as follows:

- **Cosmopedia v2:** An advanced version of the largest synthetic dataset for pretraining, comprising over 39 million generated samples from textbooks, blog posts, and stories.
- **FineWeb-Edu:** Consists of 1.3 trillion tokens of educational web pages filtered from the FineWeb dataset.

Evaluation Recent long-context evaluations have focused on a 128K context length (Zhang et al., 2024c; Hsieh et al., 2024; Yen et al., 2024b), leading to the creation of various evaluation datasets. Accordingly, we set the target context length to 128K for comprehensive evaluation. We evaluate the models using the HELMET (Yen et al., 2024b) and RULER (Hsieh et al., 2024) benchmarks. The evaluation spans five task types from the HELMET benchmark: synthetic recall, retrieval-augmented generation (RAG), many-shot in-context learning (ICL), passage re-ranking, and long-document QA, covering a total of 17 subtasks. Detailed descriptions of the HELMET benchmarks can be found in Appendix B.3. Additionally, the RULER benchmark includes 13 synthesis subtasks.

4.2 Comparison with Other Data Synthesis Methods

We first compare NExtLong with previous long-context data synthesis methods on the 128K context length setting.

Experimental Settings for Extending Context Length to 128K. We fine-tune the Meta-Llama-3-8B-base (Meta, 2024) model using a batch size of 4M tokens for 1000 steps with the open-source framework GPT-NeoX[2]. The RoPE frequency base is increased from 500,000 in Meta-Llama-3-8B-base to 200,000,000. The same training configuration is applied to all methods for a fair comparison. Further details are available in Appendix B.1.

Baseline Methods We compare NExtLong with several methods that synthesized 32,000 128K-length samples (approximately 4 billion training tokens) from short documents:

- **Standard Method:** Randomly samples and concatenates short documents (Ouyang et al., 2022; Le Scao et al., 2023; Touvron et al., 2023a).
- **KNN (Guu et al., 2020; Levine et al., 2021):** Pairs each document with the top k most similar retrieved documents.
- **ICLM (Shi et al., 2023b):** Uses a traveling salesman algorithm to reduce redundancy and improve diversity.

[2] https://github.com/EleutherAI/gpt-neox

Table 1: Comparing NExtLong with other data synthesis methods on HELMET and RULER benchmark. All results are averaged over sequence lengths of 8K,16K,32K,64K, and 128K. ♢: results from Yen et al. (2024b);♣: results evaluated by ourselves.

Model	Max Len	Avg.	Recall	RAG	ICL	Re-rank	LongQA	RULER
Meta-Llama-3-8B-base ♢	8K	13.37	18.00	12.68	13.60	7.74	10.38	17.80
+ *Standard* ♣	128K	52.85	62.33	58.67	71.24	19.18	28.99	76.68
+ KNN ♣	128K	50.97	64.24	56.00	60.28	18.77	32.27	74.30
+ ICLM ♣	128K	50.37	64.04	54.48	72.36	14.04	28.17	69.14
+ Quest ♣	128K	55.25	69.13	57.47	72.08	22.35	33.82	76.63
+ NExtLong (ours) ♣	128K	**62.58**	**82.56**	**60.91**	**81.76**	**31.47**	**37.30**	**81.50**

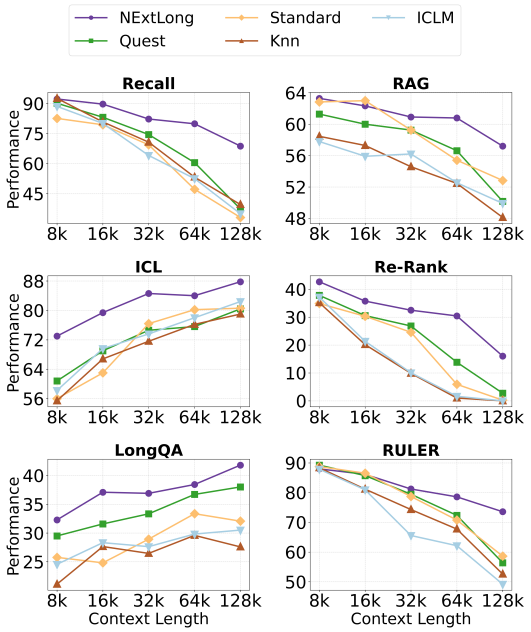

Figure 3: Comparison of NExtLong with other data synthesis methods on HELMET and RULER benchmarks across different context lengths. NExtLong shows significant performance improvements across various tasks.

- **Quest (Gao et al., 2024a):** Balances semantic correlation and diversity by clustering documents based on predicted queries.

NExtLong Outperforms Existing Data Synthesis Methods. Table 1 and Figure 3 compare NExtLong with other data synthesis methods across different context lengths (8K, 16K, 32K, 64K, and 128K) on the HELMET and RULER benchmarks. Table 1 presents the averaged results, indicating that NExtLong surpasses all baseline methods with an average improvement of at least +7.33%. Notably, it achieves a 13.43% gain in Recall and a 9.12% improvement in Re-Rank over the Quest method, demonstrating its effectiveness in enhancing long-context performance.

Figure 3 further illustrates that NExtLong outperforms other methods across varying context lengths, with the gap widening as context length increases. The results highlight NExtLong's superior capability to model long-range dependencies and maintain robust performance even at 128K context length, demonstrating the versatility and reliability of NExtLong across diverse tasks and its effectiveness in handling ultra-long contexts.

4.3 Comparison with SOTA Models

We compare NExtLong-trained models with state-of-the-art models, including ProLong (Gao et al., 2024b), which uses a two-stage training strategy: first training on shorter contexts, then extending to longer ones (e.g., 512K). ProLong evaluates models using a "train-long, test-short" approach, testing on shorter contexts (e.g., 128K). For fairness, we adopt the same strategy, training up to 512K and evaluating on 128K benchmarks.

Experimental Settings for Extending Context Length to 512K. Unlike other models such as ProLong (Gao et al., 2024b), which are trained on naturally occurring long documents, we utilize NExtLong-synthesized data for training. Specifically, we synthesize two long-context datasets, NExtLong-64K and NExtLong-512K, both derived from the FineWeb-Edu and Cosmopedia v2 corpora. The detailed training hyper-parameters are provided in Appendix B.2.

Baseline Models We select open-source base models with comparable parameter sizes for evaluation, including Llama-3.1-8B and Llama-3-8B-ProLong-512K-Base. Additionally, we compare against current SOTA closed-source models, such as GPT-4o, Gemini, and Claude.

Without Using Long Documents, NExtLong Outperforms Other Open-Source Models. Table 2 shows that Llama-3-8B-NExtLong-512K-Base model surpasses other open-source models, outperforming Llama-3-8B-ProLong-512K-Base by +5.42% and Llama-3.1-8B by +4.69% on av-

Table 2: Comparing NExtLong with other open-source base models on the HELMET and RULER benchmarks. All results are averaged over sequence lengths of 8K, 16K, 32K, 64K, and 128K. ♢: results from Yen et al. (2024b); ♣: results evaluated by ourselves.

Model	Max Len	Avg.	Recall	RAG	ICL	Re-rank	LongQA	RULER
Open-source base models								
Yarn-Llama-2-7b-128K ♣	128K	35.61	18.58	43.47	71.32	13.27	25.91	41.14
DeepSeek-V2-Lite ♣	160K	42.62	37.00	46.93	72.36	14.31	29.97	55.17
Yi-9B-200K ♣	200K	53.91	65.88	57.31	62.36	22.86	**39.47**	75.61
Qwen2.5-7B ♣	128K	49.53	59.04	49.44	73.84	18.37	28.62	67.84
Mistral-Nemo-Base ♣	128K	50.34	57.04	54.73	74.68	18.98	35.53	61.08
Llama-3-8B-ProLong-512K-Base ♣	512K	60.34	86.95	60.93	79.20	**31.66**	24.68	78.60
Llama-3.1-8B ♣	128K	61.07	83.18	61.22	71.40	29.10	37.35	84.20
Llama-3-8B-NExtLong-512K-Base (ours) ♣	512K	**65.76**	**91.58**	**63.68**	**84.08**	31.27	38.42	**85.52**
Closed-source models								
GPT-4o-mini ♢	128K	70.98	94.90	69.64	78.12	52.30	43.13	87.82
GPT-4o ♢	128K	77.43	98.22	72.30	85.76	64.70	47.61	95.96
Gemini-1.5-Pro ♢	2M	71.71	84.60	72.06	78.74	69.04	45.99	79.84
Claude-3.5-sonnet ♢	200K	51.19	93.30	41.10	59.80	9.10	10.83	93.00

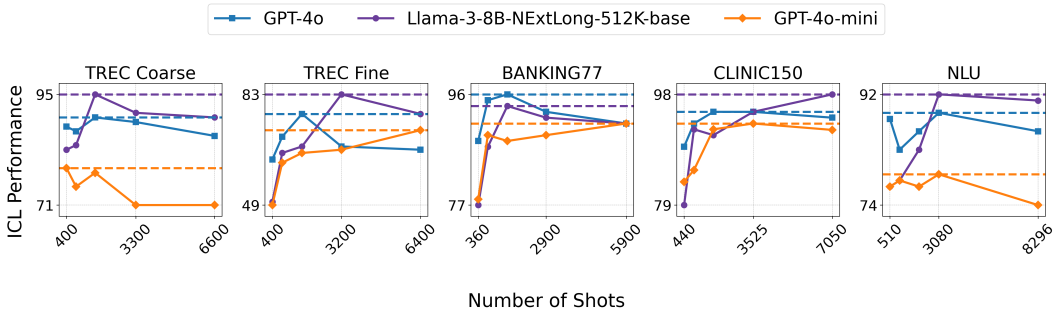

Figure 4: Comparison of NExtLong with GPT-4o on five In-Context Learning (ICL) tasks from the HELMET benchmark. Each polyline represents the model's performance across context lengths of 8K, 16K, 32K, 64K, and 128K.

erage. These results demonstrate that synthesized data can match or even surpass non-synthesized long documents in enhancing long-context capabilities, positioning NExtLong for ultralong context extensions.

In ICL Tasks, NExtLong Matches or Surpasses GPT-4o as the Number of Shots Increases. Recently, Long-context models' ICL performance has garnered significant attention (Bertsch et al., 2024; Agarwal et al., 2024; Anil et al., 2024). Agarwal et al. (2024) highlight ICL performance as a valuable metric for evaluating long-context models. Figure 4 shows that as the number of shots increases, NExtLong matches GPT-4o in the Banking77 task and outperforms it in four other tasks. Its strong performance and moderate computational cost make NExtLong suitable for future ICL applications.

5 Analysis

This section provides an in-depth analysis of the NExtLong method. Due to the high computational cost of experiments, ablation studies are conducted with a 128K context length.

5.1 NExtLong Enhances Long-Range Dependency Modeling

To assess the improvement in long-range dependency modeling achieved by NExtLong's negative document extension, we conduct a probing experiment using the Longbook QA dataset (Zhang et al., 2024b), which features long-range dependencies up to 128K in length. In this experiment, we use the normalized attention weights assigned to the first third of the context, when predicting the last token, as a metric for evaluating the model's long-dependency modeling ability.

As shown in Figure 5, we observe a positive correlation between this long-dependency metric and the model's performance on LongQA. Complementarily, as discussed in Appendix A.4, NExtLong reduces the model's dependence on proximal text (the last third context). These findings demonstrate that models trained with NExtLong's negative document extension exhibit enhanced long-dependency modeling capabilities, resulting in significantly improved long-context performance.

Table 3: Comparing NExtLong with ProLong models on the LongBench v2 benchmark.

Model	Overall	Easy	Hard	Short	Medium	Long
Llama-3-8B-ProLong-512K-Instruct	27.2	31.8	24.4	31.7	29.3	15.7
Llama-3-8B-NExtLong-512K-Instruct	**30.4**	**33.3**	**28.6**	**32.2**	**30.7**	**26.9**

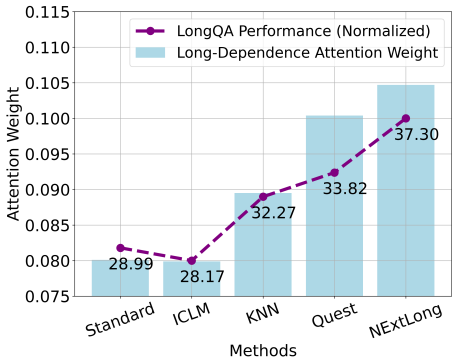

Figure 5: NExtLong enhances long-range dependency modeling. The bars represent the model's ability to capture long-range dependencies, measured by the attention weights assigned to the first third of the context. The dotted line indicates the model's performance, demonstrating a positive correlation between improved long-range dependency modeling and better performance on the LongQA task.

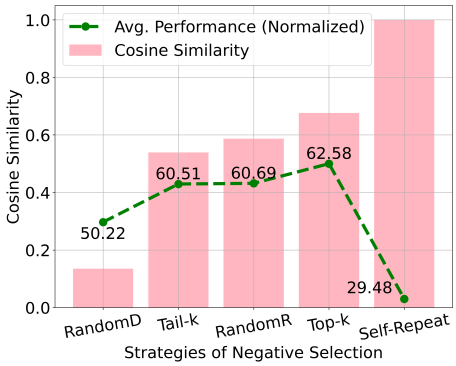

Figure 6: The impact of negative selection on long-context performance. The bars represent the cosine similarity of documents concatenated by different strategies. The dotted line indicates the average performance on the HELMET and RULER benchmarks, with all results normalized to align within the specified similarity range.

5.2 NExtLong Performs Strongly After Supervised Finetuning.

To evaluate how NExtLong performs after supervised fine-tuning, we follow the approach in ProLong (Gao et al., 2024b) and fine-tune our base model using the UltraChat (Ding et al., 2023) short-context SFT dataset. We test the model on the recently proposed LongBench v2 benchmark (Bai et al., 2024). As shown in Table 3, NExtLong outperforms ProLong overall, especially on the Long metric. We also compare NExtLong with other SOTA models in Appendix A.6. The results demonstrate that Llama-3-8B-NExtLong-512K-Base performs strongly as a base model. With the same SFT dataset, the improved long-context base model enables the training of a superior fine-tuned model.

5.3 The Importance of Hard Negatives for Achieving Better Results

To evaluate the impact of hard negatives on performance, we design 5 document retrieval strategies. For each meta-chunk, we retrieve 512 documents from the Faiss index and select k from these documents using the following strategies:

1. **Self-Repeat**: Repeat meta-chunks without including retrieved documents.
2. **Top-k (hard negatives)**: Concatenate documents in descending order of similarity until the target length is reached.
3. **RandomR**: Shuffle retrieved documents randomly and select from them.
4. **Tail-k**: Concatenate documents in ascending order of similarity.
5. **RandomD**: Randomly select documents from the training dataset, ignoring retrieval documents, which share a similar idea to (Tian et al., 2024).

Figure 6 shows that the choice of hard negatives (the **top-k** setting) plays a crucial role in NExtLong. Low-similarity negatives reduce training difficulty, weakening performance. Meanwhile, using repeated meta-chunks brings false negatives and further degrades model performance, which is consistent with the phenomenon observed in contrastive learning (Chen et al., 2021).

5.4 NExtLong Shows No Significant Performance Degradation on Short Text.

To verify how well NExtLong maintains model performance on short text tasks, following Quest (Gao et al., 2024a), we select 7 widely-used short-text datasets: HellaSwag (Hel.) (Zellers et al., 2019),

Table 4: Comparison of short text performance across methods. Overall, NExtLong shows a minor performance fluctuation in the short text benchmark as the method improves the long-context ability of Meta-Llama-3-8B-base.

Model	Avg.	Hel.	Lam.	AR-C.	AR-E.	PIQA	Win.	Log.
Meta-Llama-3-8B-base	63.75	60.13	75.66	50.34	80.18	79.60	72.85	27.50
+ *Standard*	63.66	59.57	72.87	49.83	81.73	80.36	73.64	27.65
+ KNN	63.23	59.74	72.25	49.15	80.85	80.30	73.56	26.73
+ ICLM	63.89	61.24	71.67	52.47	81.73	80.14	73.56	26.42
+ Quest	63.96	59.72	73.20	50.68	81.14	80.41	74.74	27.80
+ NExtLong	63.83	60.32	72.06	51.45	82.03	79.87	73.09	27.96

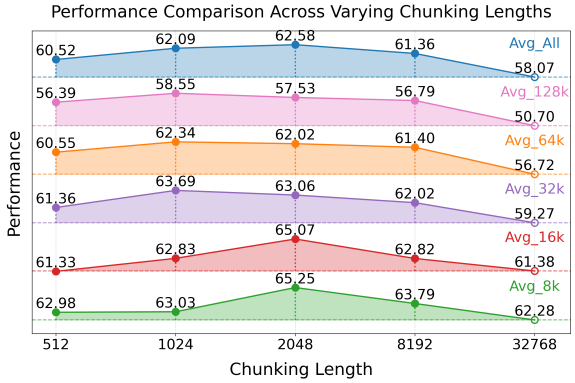

Figure 7: The impact of different chunking granularities S on the performance of NExtLong. The six curves, from bottom to top, correspond to the average performance across six task types at document lengths of 8K, 16K, 32K, 64K, and 128K, as well as the overall average across all these lengths.

Lambada_OpenAI (Lam.) (Paperno et al., 2016), ARC-Challenge (AR-C.) (Clark et al., 2018), ARC-Easy (AR-E.), PIQA (Bisk et al., 2020), WinoGrande (Win.) (Sakaguchi et al., 2021), and Logiqa (Log.) (Liu et al., 2020).

As shown in Table 4, compared to the Meta-Llama-3-8B-base model, the performance on short text evaluations shows no significant degradation after continued training with synthesized data derived from the NExtLong method.

5.5 The Impact of Chunking Granularity s

We perform an ablation study on chunking granularity s using values of 512, 1024, 2048, 8192, and 32768. The results, shown in Figure 7, indicate that the model performs best with a granularity of 2048. While a granularity of 1024 yields optimal performance for 128K context length, it underperforms in the 8k and 16k ranges compared to 2048. We conclude that too small a granularity disrupts semantic integrity, while too large introduces redundant information, negatively impacting the hard negative mining stage. A moderate granularity offers the best balance for performance.

6 Conclusion and Future Works

This paper introduces **NExtLong**, a framework that improves long-range dependency modeling in LLMs through negative document extension. By dividing a document into meta-chunks and inserting hard negative distractors, NExtLong increases learning difficulty, encouraging the LLMs to better model long-range dependencies over extended contexts. Experimental results show that NExtLong outperforms existing methods on HELMET and RULER benchmarks, achieving notable performance gains.

In the future, we plan to explore more effective negative chunk mining strategies, such as generative approaches to creating more diverse and harder distractors, further enhancing the model's ability to learn fine-grained long-range dependencies.

References

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

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

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai. *Preprint*, arXiv:2403.04652.

Ibrahim M Alabdulmohsin, Behnam Neyshabur, and Xiaohua Zhai. 2022. Revisiting neural scaling laws in language and vision. *Advances in Neural Information Processing Systems*, 35:22300–22312.

Chenxin An, Fei Huang, Jun Zhang, Shansan Gong, Xipeng Qiu, Chang Zhou, and Lingpeng Kong. 2024. Training-free long-context scaling of large language models. *arXiv preprint arXiv:2402.17463*.

Cem Anil, Esin Durmus, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Nina Rimsky, Meg Tong, Jesse Mu, Daniel Ford, et al. 2024. Many-shot jailbreaking. *Anthropic, April*.

N. N. Author. 2021. Suppressed for anonymity.

Artem Babenko and Victor Lempitsky. 2014. The inverted multi-index. *IEEE transactions on pattern analysis and machine intelligence*, 37(6):1247–1260.

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

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*.

Ramakrishna Bairi, Atharv Sonwane, Aditya Kanade, VageeshD C, Arun Iyer, Suresh Parthasarathy, Sriram Rajamani, B. Ashok, and Shashank Shet. 2023. Codeplan: Repository-level coding using llms and planning.

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. 2024. Smollm-corpus.

Yoshua Bengio and Yann LeCun. 2007. Scaling learning algorithms towards AI. In *Large Scale Kernel Machines*. MIT Press.

Amanda Bertsch, Maor Ivgi, Uri Alon, Jonathan Berant, Matthew R Gormley, and Graham Neubig. 2024. In-context learning with long-context models: An in-depth exploration. *arXiv preprint arXiv:2405.00200*.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

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

Avi Caciularu, MatthewE. Peters, Jacob Goldberger, Ido Dagan, and Arman Cohan. 2023. Peek across: Improving multi-document modeling via cross-document question-answering.

Xunxin Cai, Meng Xiao, Zhiyuan Ning, and Yuanchun Zhou. 2023. Resolving the imbalance issue in hierarchical disciplinary topic inference via llm-based data augmentation. In *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1424–1429. IEEE.

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

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

Tsai-Shien Chen, Wei-Chih Hung, Hung-Yu Tseng, Shao-Yi Chien, and Ming-Hsuan Yang. 2021. Incremental false negative detection for contrastive learning. *arXiv preprint arXiv:2106.03719*.

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

Alexis Chevalier, Jiayi Geng, Alexander Wettig, Howard Chen, Sebastian Mizera, Toni Annala, Max Jameson Aragon, Arturo Rodríguez Fanlo, Simon Frieder, Simon Machado, et al. 2024. Language models as science tutors. *arXiv preprint arXiv:2402.11111*.

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

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*.

DeepSeek-AI. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *Preprint*, arXiv:2405.04434.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*.

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*.

R. O. Duda, P. E. Hart, and D. G. Stork. 2000. *Pattern Classification*, 2nd edition. John Wiley and Sons.

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

Chaochen Gao, Xing Wu, Qi Fu, and Songlin Hu. 2024a. Quest: Query-centric data synthesis approach for long-context scaling of large language model. *arXiv preprint arXiv:2405.19846*.

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

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. 2024b. How to train long-context language models (effectively).

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, et al. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*, volume 1. MIT Press.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

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

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*.

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. 2020. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*.

Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

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

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A Smith, Iz Beltagy, et al. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *arXiv preprint arXiv:2311.10702*.

Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan O Arik. 2024a. Long-context llms meet rag: Overcoming challenges for long inputs in rag. *arXiv preprint arXiv:2410.05983*.

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

Vincent Jung and Lonneke van der Plas. 2024. Understanding the effects of language-specific class imbalance in multilingual fine-tuning. *arXiv preprint arXiv:2402.13016*.

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. 2020. Hard negative mixing for contrastive learning. *Advances in neural information processing systems*, 33:21798–21809.

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*.

Raghav Kaushik, Rajasekar Krishnamurthy, Jeffrey F Naughton, and Raghu Ramakrishnan. 2004. On the integration of structure indexes and inverted lists. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 779–790.

M. J. Kearns. 1989. *Computational Complexity of Machine Learning*. Ph.D. thesis, Department of Computer Science, Harvard University.

Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Carlos Muñoz Ferrandis, Yacine Jernite, Margaret Mitchell, Sean Hughes, Thomas Wolf, et al. 2022. The stack: 3 tb of permissively licensed source code. *arXiv preprint arXiv:2211.15533*.

P. Langley. 2000. Crafting papers on machine learning. In *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pages 1207–1216, Stanford, CA. Morgan Kaufmann.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Yoav Levine, Noam Wies, Daniel Jannai, Dan Navon, Yedid Hoshen, and Amnon Shashua. 2021. The inductive bias of in-context learning: Rethinking pretraining example design. *arXiv preprint arXiv:2110.04541*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Shanda Li, Chong You, Guru Guruganesh, Joshua Ainslie, Santiago Ontanon, Manzil Zaheer, Sumit Sanghai, Yiming Yang, Sanjiv Kumar, and Srinadh Bhojanapalli. 2023. Functional interpolation for relative positions improves long context transformers. *arXiv preprint arXiv:2310.04418*.

Hao Liu, Wilson Yan, Matei Zaharia, and Pieter Abbeel. 2024. World model on million-length video and language with ringattention. *arXiv preprint arXiv:2402.08268*.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. 2024. Fineweb-edu.

Antonio Mallia, Omar Khattab, Torsten Suel, and Nicola Tonellotto. 2021. Learning passage impacts for inverted indexes. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1723–1727.

Sahisnu Mazumder and Bing Liu. 2022. Lifelong and continual learning dialogue systems.

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date.

R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors. 1983. *Machine Learning: An Artificial Intelligence Approach, Vol. I*. Tioga, Palo Alto, CA.

T. M. Mitchell. 1980. The need for biases in learning generalizations. Technical report, Computer Science Department, Rutgers University, New Brunswick, MA.

A. Newell and P. S. Rosenbloom. 1981. Mechanisms of skill acquisition and the law of practice. In J. R. Anderson, editor, *Cognitive Skills and Their Acquisition*, chapter 1, pages 1–51. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ.

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. 2023. Openwebmath: An open dataset of high-quality mathematical web text. *arXiv preprint arXiv:2310.06786*.

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

Alec Radford. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2020. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*.

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

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. 2019. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*.

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

A. L. Samuel. 1959. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):211–229.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023a. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Victoria Lin, Noah A Smith, Luke Zettlemoyer, Scott Yih, and Mike Lewis. 2023b. In-context pretraining: Language modeling beyond document boundaries. *arXiv preprint arXiv:2310.10638*.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama.

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*.

Hui Su, Zhi Tian, Xiaoyu Shen, and Xunliang Cai. 2024. Unraveling the mystery of scaling laws: Part i. *arXiv preprint arXiv:2403.06563*.

Jianlin Su. 2023. Rectified rotary position embeddings. https://github.com/bojone/rerope.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *Preprint*, arXiv:2104.09864.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Junfeng Tian, Da Zheng, Yang Cheng, Rui Wang, Colin Zhang, and Debing Zhang. 2024. Untie the knots: An efficient data augmentation strategy for long-context pre-training in language models. *arXiv preprint arXiv:2409.04774*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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

Szymon Tworkowski, Konrad Staniszewski, Mikołaj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Miłoś. 2024. Focused transformer: Contrastive training for context scaling. *Advances in Neural Information Processing Systems*, 36.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):1–26.

Xing Wu, Guangyuan Ma, Meng Lin, Zijia Lin, Zhongyuan Wang, and Songlin Hu. 2023a. Contextual masked auto-encoder for dense passage retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4738–4746.

Xing Wu, Guangyuan Ma, Wanhui Qian, Zijia Lin, and Songlin Hu. 2022. Query-as-context pretraining for dense passage retrieval. *arXiv preprint arXiv:2212.09598*.

Yunshu Wu, Hayate Iso, Pouya Pezeshkpour, Nikita Bhutani, and Estevam Hruschka. 2023b. Less is more for long document summary evaluation by llms. *arXiv preprint arXiv:2309.07382*.

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

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. 2024. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

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*.

Yizhe Xiong, Xiansheng Chen, Xin Ye, Hui Chen, Zijia Lin, Haoran Lian, Jianwei Niu, and Guiguang Ding. 2024. Temporal scaling law for large language models. *arXiv preprint arXiv:2404.17785*.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2024. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *arXiv preprint arXiv:2406.08464*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Michihiro Yasunaga, Armen Aghajanyan, Weijia Shi, Rich James, Jure Leskovec, Percy Liang, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2022. Retrieval-augmented multimodal language modeling. *arXiv preprint arXiv:2211.12561*.

Howard Yen, Tianyu Gao, and Danqi Chen. 2024a. Long-context language modeling with parallel context encoding. *arXiv preprint arXiv:2402.16617*.

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

Haofei Yu, Yue Zhang, Wei Bi, et al. 2023. Trams: Training-free memory selection for long-range language modeling. *arXiv preprint arXiv:2310.15494*.

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*.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1503–1512.

Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. 2024a. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*.

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

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, et al. 2024c. ∞bench: Extending long context evaluation beyond 100k tokens. *arXiv preprint arXiv:2402.13718*.

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 More Ablations

A.1 Placing Meta-Chunk at Different Positions

We explore three different strategies for combining meta-chunk and hard negatives, which are represented by the following descriptions:

1. **Head**: Placing the meta-chunk at the beginning of the retrieved hard negatives.
2. **Tail**: Placing the meta-chunk at the end of the retrieved hard negatives.
3. **Random**: Randomly inserting the meta-chunk within the retrieved hard negatives.

Table 5 shows that placing the meta-chunk at the beginning (Head) yields better performance. We believe this method helps establish longer dependencies, resulting in enhanced effectiveness.

A.2 Document Length Distribution of Cosmopedia V2 and FineWebEdu

We analyze the document length distribution of two datasets, Cosmopedia V2 and FineWebEdu, by sampling 8 million documents from each dataset and encoding them using the Meta-Llama-3-8B tokenizer. Document lengths are categorized into two ranges: $[0, 8192]$ and > 8192. Table 6 shows that the majority of documents in both datasets are relatively short (under 8K). We apply the NExtLong algorithm to extend the document length to 128K and 512K, achieving approximately a 64-fold increase compared to the original.

Table 6: Document Length Distribution of Cosmopedia V2 and FineWebEdu.

Dataset	$0 \leq$ Length ≤ 8192	Length > 8192
Cosmopedia V2	100.00%	0.00%
FineWebEdu	99.19%	0.81%

Table 5: Performance Comparison of Different Insertion Strategies.

Model	Avg.	Recall	RAG	ICL	Re-rank	LongQA	RULER
Head	**62.58**	**82.56**	60.91	**81.76**	31.47	**37.30**	**81.50**
Tail	60.01	72.66	**63.67**	80.68	30.73	34.09	78.26
Random	58.95	74.51	63.05	71.64	**32.78**	33.00	78.73

A.3 Dataset Ablation Study

We compared three different dataset selection strategies: (1) using FineWeb-Edu alone for long-context data synthesis, (2) using Cosmopedia v2 alone for long-context data synthesis, and (3) combining both datasets for long-context data synthesis. The results are shown in Table 7. The findings indicate that the combined strategy achieved the best performance, highlighting that a diverse dataset significantly enhances data synthesis.

A.4 NExtLong Reduces Dependence on Proximal Text

Complementary with Section 5.1, we investigated the dependence of different models on proximal text (the last third of the text). As shown in Figure 8, NExtLong demonstrates a lower degree of dependence on proximal text. This shift in attention toward long-range text contributes to improving the model's performance.

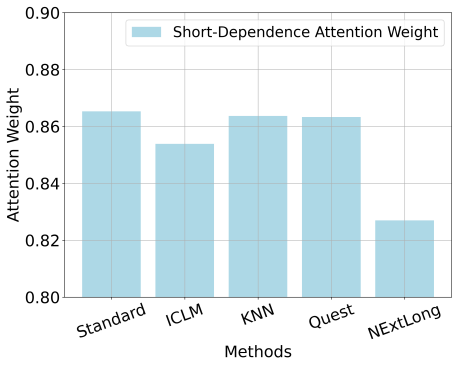

Figure 8: NExtLong reduces the model's dependence on proximal text. We calculated the degree of dependence on proximal text (the last third of the text) for different methods when completing the LongQA task. It can be observed that NExtLong significantly reduces the model's dependence on proximal text.

A.5 The Result on Needle-in-a-Haystack Benchmark

Following previous works (Gao et al., 2024a; Zhang et al., 2024a; Liu et al., 2024), we evaluate the Llama-3-8B-NExtLong-512K-Base model on the widely used Needle-in-a-Haystack task. As shown in Figure 9, Llama-3-8B-NExtLong-512K-Base achieves a 100% accuracy on the Needle-in-a-Haystack task.

A.6 NExtLong Outperforms Current SOTA Models

To evaluate the performance of instruct supervised NExtLong against state-of-the-art (SOTA) models, we compare it with several leading instruct models, including GLM-4-9B (GLM et al., 2024), Qwen2.5-7B (Yang et al., 2024), Llama3.1-8B (Meta, 2024), and GPT-4o-mini. The comparison is based on their performance on the LongBench v2 benchmark (Bai et al., 2024), a comprehensive suite designed to evaluate long-context understanding. Different from Section 5.2, we fine-tune the Llama-3-8B-NExtLong-512K-Base model with the public Magpie-Llama-3.3-Pro-1M-v0.1 (Xu et al., 2024) dataset to achieve better performance.

As shown in Table 8, NExtLong achieves the highest overall performance with 30.8%, outperforming GLM-4-9B by +0.6%, Qwen2.5-7B by +0.8%, Llama3.1-8B by +0.8%, and GPT-4o-mini by +1.5%. Notably, NExtLong demonstrates a significant improvement over models with similar parameter counts. This indicates the effectiveness of NExtLong's training methodology in handling long-context tasks.

B Experiment Details

B.1 Training Llama-3-8B-NExtLong-128K detailed setup

We use the parameters listed in Table 9 to train the 128K model. For other data synthesis methods, we only modify the training dataset while keeping all other training parameters unchanged.

B.2 Training Llama-3-8B-NExtLong-512K-Base detailed setup

We use the parameters listed in Table 10 to train the 512K model. The training samples are sourced from the NExtLong-512K and NExtLong-64k datasets in a ratio of 1:2.

Table 7: Dataset Comparison on HELMET and RULER Benchmark.

Dataset	Avg.	Recall	RAG	ICL	Re-rank	LongQA	RULER
Cosmopedia	59.26	79.94	59.58	81.00	27.07	29.13	78.84
FineWebEdu	62.04	**83.56**	**61.53**	**83.76**	28.31	34.46	80.60
Cosmopedia + FineWebEdu	**62.58**	82.56	60.91	81.76	**31.47**	**37.30**	**81.50**

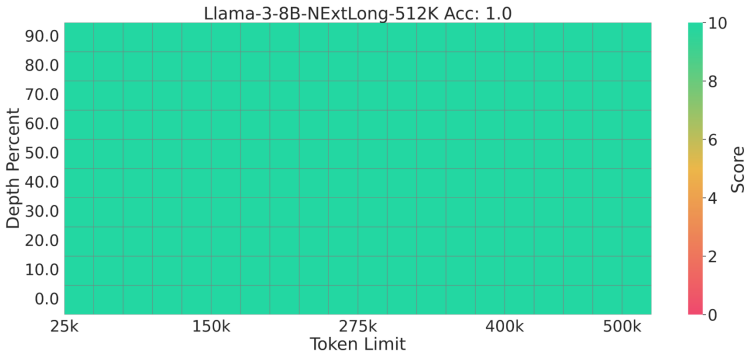

Figure 9: The Needle-in-a-Haystack task assesses a model's capability to extract specific information (the needle) from a large corpus of documents (the haystack). The y-axis indicates the position of the "needle" within the document, spanning from 25K to 500K tokens.

Table 8: Comparison of NExtLong with current SOTA models on LongBench v2 benchmark.

Model	Params	Overall (%)
NExtLong	8B	**30.8**
GLM-4-9B	9B	30.2
Qwen2.5-7B	7B	30.0
Llama3.1-8B	8B	30.0
GPT-4o-mini	-	29.3

B.3 Evaluation Metric and Task Category of the HELMET Benchmark.

The task category and evaluation metric of the HELMET benchmark (Yen et al., 2024b), which we use in this work, are shown in Table 11.

C More Details in NExtLong

C.1 The Calculation Method for the Number of Hard Negatives k

In the Negative Document Extension stage, the meta-document targeted for extension is chunked into meta-chunks. Each meta-chunk retrieves the top-k similar texts as hard negatives from the Faiss index, with the value of k adaptively adjusted based on the target length T. Let E represent the encoding rate of the model tokenizer. The total number of characters Q needed for retrieval is calculated as follows:

$$Q = T \times E \times w \tag{8}$$

Here, the adjustment factor w accounts for variability and ensures that a sufficient number of hard negatives are recalled for each meta-chunk. In our experiments, we set $w = 1.5$. Then, we compute the remaining characters L necessary for synthesis beyond the total character length S of the meta-document. This is calculated by subtracting S from Q:

$$L = Q - S \tag{9}$$

The meta-document is divided into p meta-chunks. The number of characters P required for retrieval from each meta-chunk is distributed evenly across all p meta-chunks:

$$P = \frac{L}{p} \tag{10}$$

The number of hard negatives k that need to be retrieved for each meta-chunk can be calculated as follows:

$$k = \frac{P}{s} \tag{11}$$

Substituting P into this equation gives:

$$k = \frac{L}{p \times s} = \frac{Q - S}{p \times s} = \frac{T \times E \times w - S}{p \times s} \tag{12}$$

This formulation ensures that the number of hard negatives k is proportional to the chunking granularity s and the target length T. Additionally, to enhance content diversity, we ensure that the same hard negative is not repeatedly used across different meta-chunks.

Table 9: 128K model training configuration.

128K training setting	
Initial Model	Meta-Llama-3-8B (base model)
rotary-emb-base	200,000,000
β_1	0.9
β_2	0.95
lr	$4e^{-5}$
precision	bfloat16
gradient-clipping	1.0
weight-decay	0.1
lr-decay-style	cosine
train-iters	1000
warmup-iters	200
seq-length	131072
GPU-type	H100
GPU-numbers	64
training-time	15h

Table 10: 512K model training configuration.

512K training setting	
Initial Model	Llama-3-8B-ProLong-64k-Base
rotary-emb-base	128,000,000
β_1	0.9
β_2	0.95
lr	$1e^{-5}$
precision	bfloat16
gradient-clipping	1.0
weight-decay	0.1
lr-decay-style	cosine
train-iters	500
warmup-iters	50
seq-length	524288
GPU-type	H100
GPU-numbers	128
training-time	20h

C.2 Pseudocode of NExtLong

We present the complete process of constructing the NExtLong dataset using pseudocode in Algorithm 1.

Table 11: Summary of datasets and metrics in HELMET benchmark.

Category	Dataset	Metrics	Description
Retrieval-augmented generation	Natural Questions	SubEM	Factoid question answering
	TriviaQA	SubEM	Trivia question answering
	PopQA	SubEM	Long-tail entity question answering
	HotpotQA	SubEM	Multi-hop question answering
Passage re-ranking	MS MARCO	NDCC@10	Rerank passage for a query
Long-document QA	NarrativeQA	ROUGE F1	Book and movie script QA
	∞BENCH QA	ROUGE F1	Novel QA with entity replacement
	∞BENCH MC	Accuracy	Novel multiple-choice QA with entity replacement
Many-shot in-context learning	TREC Coarse	Accuracy	Question type classification, 6 labels
	TREC Fine	Accuracy	Question type classification, 50 labels
	NLU	Accuracy	Task intent classification, 68 labels
	BANKING77	Accuracy	Banking intent classification, 77 labels
	CLINC150	Accuracy	Intent classification, 151 labels
Synthetic recall	JSON KV	SubEM	Retrieve a key in JSON dictionary
	RULER MK Needle	SubEM	Retrieve the needle (a number) within noisy needles
	RULER MK UUID	SubEM	Retrieve the needle (a UUID) within noisy needles
	RULER MV	SubEM	Retrieve multiple values for one needle (key)

Algorithm 1 Negative Document Extension (NExtLong)

Input: Training corpus $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$, chunking granularity s, number of retrieved hard negatives k, target long length L_{target}, Faiss index *Faiss*
Output: Synthesized long documents for long-dependence modeling

Initialize an empty list $\mathcal{T}$ for storing synthesized long documents

function DOCUMENT_CHUNKING(r, s)
  Split r into paragraphs $r = \{r_1, r_2, \ldots, r_P\}$ by newline characters
  Initialize an empty list chunks $= []$
  Initialize an empty buffer buffer $= []$ with length counter $\ell = 0$
  for each paragraph r_i in r **do**
    if $\ell +$ Length$(r_i) \leq s$ **then**
      buffer $\leftarrow$ buffer $\cup\, r_i$
      $\ell \leftarrow \ell +$ Length(r_i)
    else
      chunks $\leftarrow$ chunks $\cup$ {buffer}
      buffer $\leftarrow r_i$
      $\ell \leftarrow$ Length(r_i)
    end if
  end for
  if buffer $\neq \varnothing$ **then**
    chunks $\leftarrow$ chunks $\cup$ {buffer}
  end if
  return chunks
end function

function NEGATIVE_MINING(m_i, k, *Faiss*)
  $n_{i_1}, n_{i_2}, \ldots, n_{i_k} \leftarrow$ Top-k similar chunks from *Faiss* to m_i
  return $[\, m_i, n_{i_1}, \ldots, n_{i_k}]$
end function

procedure NEXTLONG($\mathcal{D}, s, k, L_{\text{target}}$, *Faiss*)
  Build Faiss index by segmenting each $d_j \in \mathcal{D}$ into chunks
  Insert the embeddings of all chunks into *Faiss*
  for each document $r \in \mathcal{D}$ **do**
    $\{m_1, m_2, \ldots, m_p\} \leftarrow$ DOCUMENT_CHUNKING(r, s)
    for $i \leftarrow 1$ to p **do**
      $l_i \leftarrow$ NEGATIVE_MINING$(m_i, k,$ *Faiss*$)$
    end for
    $t \leftarrow [\, l_1, l_2, \ldots, l_p\,]$ ▷ Concatenate into a single long document
    if Length(t) $\geq L_{\text{target}}$ **then**
      $\mathcal{T} \leftarrow \mathcal{T} \cup \{t\}$
    end if
  end for
  return $\mathcal{T}$ ▷ Synthesized long documents for training
end procedure