

# From 128K to 4M: Efficient Training of Ultra-Long Context Large Language Models

Chejian Xu<sup>\*†1</sup>, Wei Ping<sup>†2</sup>, Peng Xu<sup>2</sup>, Zihan Liu<sup>2</sup>

Boxin Wang<sup>2</sup>, Mohammad Shoeybi<sup>2</sup>, Bo Li<sup>1</sup>, Bryan Catanzaro<sup>2</sup>

## Abstract

Long-context capabilities are essential for a wide range of applications, including document and video understanding, in-context learning, and inference-time scaling, all of which require models to process and reason over long sequences of text and multimodal data. In this work, we introduce an efficient training recipe for building ultra-long context LLMs from aligned *instruct* model, pushing the boundaries of context lengths from 128K to 1M, 2M, and 4M tokens. Our approach leverages continued pretraining strategies to extend the context window, while employing efficient instruction tuning to maintain short context capabilities. Our **UltraLong-8B**, built on Llama-3.1-Instruct, achieves state-of-the-art performance across a diverse set of long-context benchmarks. Importantly, UltraLong-8B also maintains competitive performance on standard benchmarks, showing balanced improvements for both long and short context tasks. We provide an in-depth analysis of key design choices, highlighting the impacts of scaling strategies and data composition. Our findings establish a robust framework for efficiently scaling context lengths while preserving general model capabilities. We released all model weights for open research. We release all model weights at: <https://research.nvidia.com/labs/adlr/ultralong/>.

## 1 Introduction

Large language models (LLMs) have demonstrated remarkable performance across a diverse range of text and multimodal tasks (Hurst et al., 2024; Gemini et al., 2024; Liu et al., 2024a; Yang et al., 2024; Dai et al., 2024). However, many applications, such as document and video understanding (Gemini et al., 2024; Azzolini et al., 2025), in-context learning, and inference-time scaling (Guo et al.,

2025), demand the ability to process and reason over extremely long sequences of tokens (Gemini et al., 2024; Xu et al., 2023; Yang et al., 2025; Xu et al., 2024b; Lu et al., 2024). In these scenarios, the limited context window of LLMs poses a significant bottleneck, as critical information scattered across lengthy documents may be overlooked. This limitation motivates the need for models that can efficiently handle ultra-long contexts without sacrificing performance on standard tasks.

Recent trends in both industry and academia have focused on extending the context windows of LLMs. Proprietary systems like GPT-4o (Hurst et al., 2024) support context lengths of up to 128K tokens, while the reasoning model o1 (Jaech et al., 2024) further pushes this limit to 200K tokens to accommodate inference-time scaling. Other models, such as Claude 3.5 Sonnet (Anthropic, 2024) and Gemini 1.5 Pro (Gemini et al., 2024), have demonstrated the feasibility of handling even larger contexts. Despite significant progress in open-access models, many efforts have been limited by the lack of detailed training data blends (Dubey et al., 2024) and efficient extension recipes (Gao et al., 2024). Moreover, evaluations of these models have often relied on synthetic benchmarks that do not fully capture their performance on real-world long-context tasks (Pekelis et al., 2024).

In this work, we present a systematic recipe for training ultra-long context language models. Our approach involves two key stages. The first stage, continued pretraining, extends the context window of LLMs to ultra-long lengths (up to 1M, 2M, and 4M tokens) by leveraging a specially curated corpus. We introduce techniques such as the use of special document separators during concatenation and apply YaRN-based (Peng et al., 2023b) RoPE scaling to improve the model’s ability to process long sequences. The second stage, instruction tuning, refines the model’s instruction-following and reasoning capabilities using a high-quality, short-

<sup>\*</sup>Work done during an internship at NVIDIA.  
<sup>1</sup>UIUC, <sup>2</sup>NVIDIA. <sup>†</sup>Correspondence to: Chejian Xu <chejian2@illinois.edu>, Wei Ping <wping@nvidia.com>.

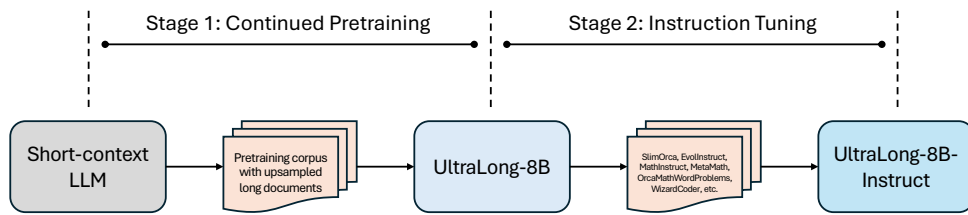


Figure 1: Overview of our training pipeline. In Stage 1, the model’s context window is extended through continued pretraining, leveraging techniques such as special document separators and YaRN-based scaling to handle ultra-long sequences. In Stage 2, instruction tuning is applied using a curated dataset to enhance the model’s instruction-following and reasoning capabilities. This pipeline enables the development of language models that achieve good performance on both long-context and standard benchmarks.

context supervised fine-tuning (SFT) dataset across general, mathematical, and coding domains.

We validate our approach using Llama-3.1-8B-Instruct with 128K context window (Dubey et al., 2024) as the starting point and conduct extensive evaluations on both synthetic and real-world long-context benchmarks, including RULER (Hsieh et al., 2024), LV-Eval (Yuan et al., 2024b), and InfiniteBench (Zhang et al., 2024). Additionally, we benchmark our models on standard datasets such as MMLU (Hendrycks et al., 2020), MATH (Hendrycks et al., 2021), GSM-8K (Cobbe et al., 2021), and HumanEval (Chen et al., 2021) to ensure that the extended context does not compromise general task performance. Evaluation results demonstrate that our final models, **UltraLong-8B**, achieve state-of-the-art (SOTA) long-context performance while maintaining competitive performance on standard benchmarks.

Our contributions are summarized as follows:

- We propose an efficient and scalable training recipe that extends the context window of LLMs to ultra-long lengths (up to 4M tokens) while preserving, and in some cases enhancing, performance on standard benchmarks.
- We introduce curated data construction strategies and context extension techniques such as the use of special document separators during data preparation and apply YaRN-based scaling for positional embeddings, both of which are shown through ablation studies to be essential for effective long-context modeling.
- We show that a one-step continued pretraining strategy is more efficient than a multi-step approach for context extension, consistently yielding superior results on both synthetic and real-world long-context benchmarks.

- We conduct extensive experiments on benchmarks including RULER, LV-Eval, InfiniteBench, MMLU, MMLU-Pro, MATH, GSM-8K, and HumanEval, showing that our models outperform existing baselines in both long-context and standard tasks.

The remainder of the paper is organized as follows. Section 2 reviews related work on long-context language modeling and extension strategies. Section 3 details our training methodology, including both the continued pretraining and instruction tuning stages. In Section 4, we describe the baseline models and evaluation benchmarks used in our experiments, while Section 5 presents our experimental results. Section 6 provides an in-depth analysis through ablation studies, and Sections 7 and 8 concludes the paper by discussing limitations and outlining directions for future research.

## 2 Related Work

### 2.1 Long-context methods

Existing context extension strategies for long-context language models can be broadly categorized into three groups: exact attention methods, approximate attention methods, and approaches that incorporate additional modules. Exact attention methods enhance the parameterization of the attention mechanism to support longer sequences. Techniques such as Position Interpolation (PI) (Chen et al., 2023b), NTK-aware (bloc97, 2023), Dynamic NTK (emozilla, 2023), YaRN (Peng et al., 2023b), and CLEX (Chen et al., 2023a)—all based on RoPE (Su et al., 2024)—design position embeddings that enable length extension. These approaches can be applied either through fine-tuning or to frozen models. In contrast, approximate attention methods adopt structured approximations to mitigate the computational cost of long-context

processing. For example, LongLoRA (Chen et al., 2023c) combines LoRA (Hu et al., 2021) with Shifted Sparse Attention to reduce overhead, while LM-Infinite (Han et al., 2024) limits attention to a few tokens at the beginning of the text and a local window to remain within the pretrained length. Other approaches, such as Dual Chunk Attention (An et al., 2024), decompose attention into chunk-based modules to better capture the relative positional information, and some works (Xu et al., 2023) leverage retrieval mechanisms to extract relevant blocks from long documents. Additionally, methods that introduce extra modules (Hwang et al., 2024; Ren et al., 2024) focus on compressing the information in the long input contexts. In this work, we focus on exact attention techniques that accurately compute full attention over extended sequences, and we introduce an efficient training recipe to enable models to handle ultra-long contexts more effectively.

## 2.2 Long-context LLMs

Recent advancements in long-context LLMs include proprietary models such as GPT-4o (Hurst et al., 2024), Gemini (Gemini et al., 2024), and Claude (Anthropic, 2024), which support extensive context windows and can process hundreds of thousands of tokens, though their closed-source nature limits reproducibility. Among open-source efforts, ProLong (Gao et al., 2024) employs NTK-aware scaling and trains on over 40B tokens, making it computationally expensive, while Gradient (Pekelis et al., 2024) uses a multi-step continued pretraining strategy that sacrifices standard task performance. ChatQA 2 (Xu et al., 2024b) develops long-context LLMs that excel in both long-context understanding and retrieval-augmented generation. In contrast, our work offers a balanced solution that extends the context window to ultra-long lengths while maintaining competitive performance on standard benchmarks through efficient continued pretraining and instruction tuning.

## 3 Method

In this section, we present our training recipe for ultra-long context models, as illustrated in Figure 1. Our approach consists of two key stages: continued pretraining and instruction tuning. Based on Llama-3.1-8B-Instruct (Dubey et al., 2024), the continued pretraining stage extends the context window of the model from 128K tokens to the target lengths (e.g.,

1M, 2M, and 4M tokens). Subsequently, the instruction tuning stage refines the model to enhance its instruction-following and reasoning abilities. Together, these stages enable our models to effectively process ultra-long inputs while maintaining strong performance on both long and short-context tasks. More details can be found in Appendix A.

### 3.1 Continued Pretraining for Context Length Extension

In the first stage, we extend the context window of Llama-3.1-8B-Instruct to the target length through continued pretraining.

**Data Preparation and Document Concatenation.** We construct our long-context pretraining corpus following the methodology outlined by Fu et al. (2024). To emphasize long-context data, we downsample documents shorter than 4K tokens and upsample those longer than 8K tokens, resulting in a corpus of 1 billion tokens. These documents are then concatenated to form longer sequences corresponding to the target context lengths (e.g., 1M, 2M, and 4M tokens). During concatenation, we separate individual documents using special characters rather than the reserved beginning and ending tokens (“<|begin\_of\_text|>” and “<|end\_of\_text|>”). Furthermore, we do not apply the cross-document attention mask (Gao et al., 2024) during continued pretraining, allowing the model to attend to the entire input sequence. Our preliminary experiments indicate that this document separation strategy, combined with full attention, enables the model to adapt more effectively and efficiently to ultra-long contexts. Detailed ablation studies and analysis are provided in Section 6.

**RoPE Scaling.** To support ultra-long context lengths, we adopt a YaRN-based scaling approach (Peng et al., 2023b) rather than the NTK-aware scaling strategies employed in previous work (Xu et al., 2024b; Gao et al., 2024; Pekelis et al., 2024). We fix the hyperparameters as  $\alpha = 1$  and  $\beta = 4$ , and compute the scale factor  $s$  based on the target context length. We observed that the Llama-3.1 model’s performance degrades when the input length approaches the maximum limit. To mitigate this, we employ a larger scaling factor for the RoPE embeddings, thereby better accommodating extended sequences.

**Implementation Details.** We build long-context models targeting three context lengths: 1M, 2M,

and 4M tokens. We set the RoPE scaling factors to  $s = 128, 256,$  and  $512$  accordingly. Each model is trained on 1B tokens for one epoch using a learning rate of  $3 \times 10^{-5}$ . For scalability, we train the models using the Megatron-LM framework (Shoeybi et al., 2019). To handle ultra-long input sequences, we adopt tensor parallelism (TP) with  $tp = 8$  and leverage context parallelism (CP) by setting  $cp = 4$  for the 1M model and  $cp = 16$  for the 2M and 4M models. Training is done on NVIDIA H100 GPUs and the full training pipeline consumed 1.25K GPU hours, which is 40× fewer than ProLong. Note that extending context length without degrading performance on short-context tasks is considered expensive in the literature. For example, DeepSeek-V3 (Liu et al., 2024a) reportedly used 119K GPU hours to scale to 128K context, which is 23.8x more than their post-training budget.

### 3.2 Instruction Tuning

In the second stage, we enhance the instruction-following and reasoning capabilities of our long-context language models through supervised fine-tuning (SFT) (Ouyang et al., 2022) on carefully curated datasets.

**Data Preparation.** We subsample a high-quality blend of SFT datasets from (Liu et al., 2024b) by integrating and refining multiple open-source SFT datasets spanning three key domains: general, mathematics, and code.<sup>1</sup> For the general domain, we incorporate data from ShareGPT (Chiang et al., 2023; The-Vicuna-Team, 2023), SlimOrca (Lian et al., 2023; Mukherjee et al., 2023), EvolInstruct (Xu et al., 2024a), GPTeacher (Teknium, 2023), AlpacaGPT4 (Peng et al., 2023a), and UltraInteract (Yuan et al., 2024a). In the math domain, our dataset includes OrcaMathWordProblems (Mitra et al., 2024), MathInstruct (Yue et al., 2023), and MetaMath (Yu et al., 2023). For the code domain, we incorporate Magicoder (Wei et al., 2024), WizardCoder (Luo et al., 2023), and GlaiveCode-Assistant (Glaive-AI, 2023).

To further enhance the quality of our SFT dataset, we leverage OpenAI’s GPT-4o (Hurst et al., 2024) and GPT-4o-mini (OpenAI, 2024) to refine the responses associated with these prompts. Finally, we perform rigorous data decontamination to ensure

<sup>1</sup>The full dataset is available for download at: <https://huggingface.co/datasets/nvidia/AceMath-Instruct-Training-Data>. We subsample our SFT data from the ‘general\_sft\_stage2’ split.

that our dataset does not include any prompts from benchmark test datasets.

Notably, our SFT blend exclusively comprises the short-context data described above, consisting of instances shorter than 8K tokens, without incorporating synthetic long-context instruction data as employed in Xu et al. (2024b); Zhao et al. (2024). We find that relying solely on short-context data is sufficient to achieve strong results in our setting, aligning with observations from prior work (Gao et al., 2024).

**Implementation Details.** We construct an SFT dataset comprising 100K examples. For every model extended to one of the three target context lengths, we use a batch size of 128 and a learning rate of  $5 \times 10^{-6}$ . Training is performed using the Megatron-LM framework (Shoeybi et al., 2019) on 256 NVIDIA H100 GPUs with tensor parallelism set to  $tp = 8$ . Each training run completes in approximately 30 minutes.

## 4 Baselines and Evaluation Benchmarks

### 4.1 Long context models

We compare our models against SOTA long context models built on the Llama family to ensure a fair and controlled evaluation of our training recipe. **Llama-3.1** (Llama-3.1-8B-Instruct) (Dubey et al., 2024) serves as our base model and features a 128K context window. **ProLong** (Llama-3-8B-ProLong-512k-Instruct) (Gao et al., 2024) is a long-context model built on Llama-3 with a 512K context window. This model is trained using two stages of continued pretraining and additional SFT, on a total of 41B tokens. **Gradient** (Llama-3-8B-Instruct-Gradient-1048k) (Pekelis et al., 2024) is another Llama-based long-context model, supporting a 1M context window. It is trained through four stages of continued pretraining on a total of 1.4B tokens, and additional SFT to strengthen its chat capabilities. Focusing exclusively on models within the Llama family allows us to clearly isolate and demonstrate the effectiveness of our training recipe for extending context lengths, while ensuring that standard task performance remains competitive.

### 4.2 Long context benchmarks

We evaluate the long-context capabilities of our models using the following benchmarks.

**RULER** (Hsieh et al., 2024) is a benchmark that assesses long-context language models by generating synthetic examples with configurable sequence

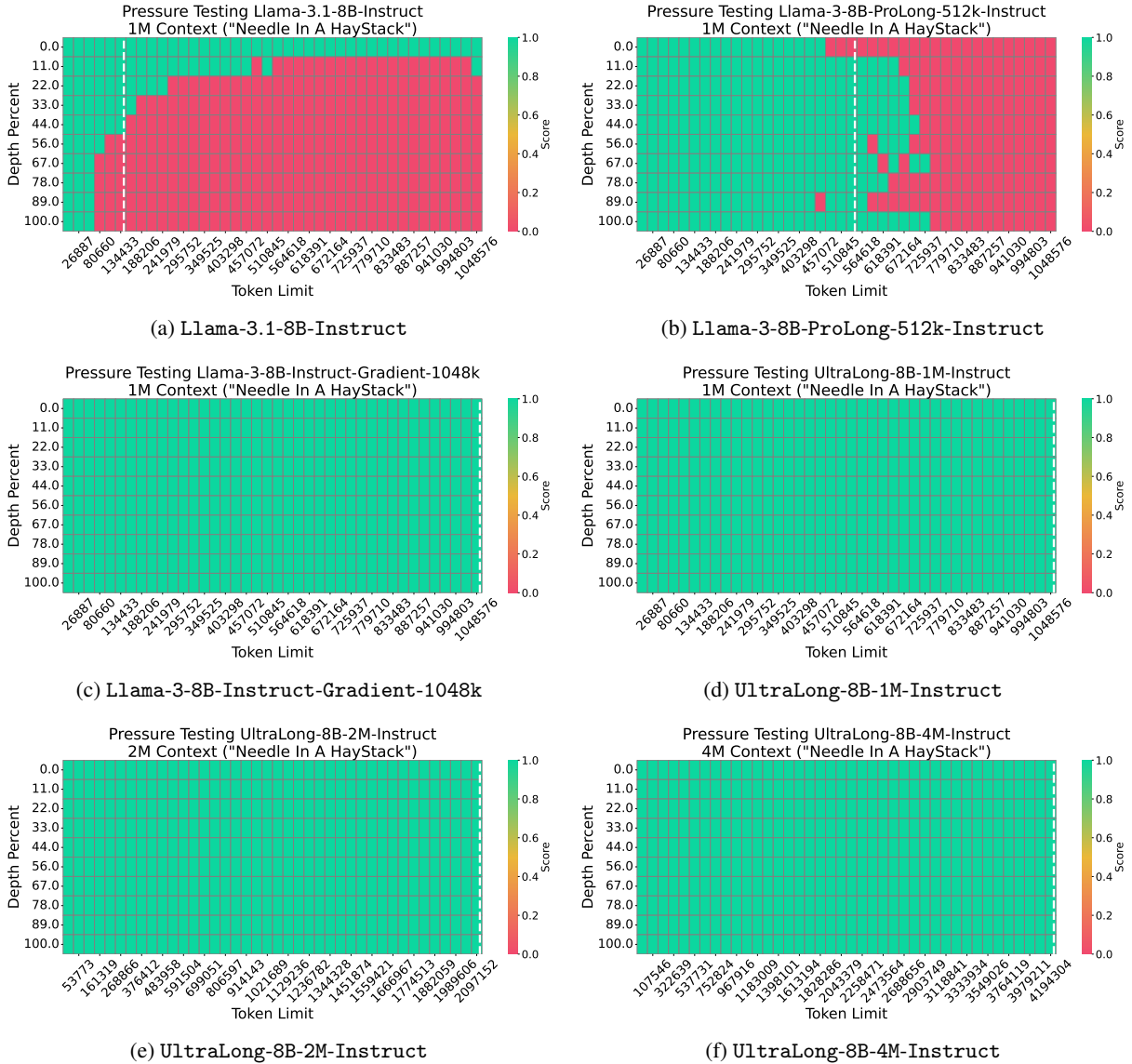


Figure 2: Needle in a Haystack passkey retrieval test results. Three baseline models are evaluated up to a 1M token context, while our models are tested at their respective maximum context lengths of 1M, 2M, and 4M tokens. Our models achieve 100% accuracy across all input lengths and depths, showing strong long-context retrieval capability.

lengths across four task categories. The benchmark originally evaluates models within a 128K context window by computing the average accuracy across different input lengths. We adopt the same generation protocol and construct test cases with lengths up to 1M tokens, including 256K, 512K, and 1M. Since some of our baseline models support input lengths only up to 128K or 512K, we report the average accuracy for inputs shorter than 128K, 512K, and 1M tokens, respectively.

**LV-Eval** (Yuan et al., 2024b) is a long-context benchmark featuring five length levels up to 256K tokens, and focuses on two primary tasks: single-hop QA and multi-hop QA. We follow the evaluation protocol and compute the average F1 score

across input lengths below 128K and 256K tokens. **InfiniteBench** (Zhang et al., 2024) is a long-context benchmark with an average input length of around 200K tokens and a maximum length exceeding 2M tokens. It includes both synthetic and real-world tasks. As this benchmark does not provide scores for specific input length levels, we follow the standard evaluation protocol and compute the average score across all tasks.

### 4.3 Standard benchmarks

We assess the standard capabilities of our models across three domains. For the general domain, we evaluate using **MMLU** (Hendrycks et al., 2020) and **MMLU-Pro** (Wang et al., 2024), where we

report the 5-shot accuracy. In the math domain, we employ two benchmarks: **MATH** (Hendrycks et al., 2021), for which we report the 0-shot exact match accuracy, and **GSM-8K** (Cobbe et al., 2021), for which we report the 8-shot exact match accuracy. For the code domain, we consider **HumanEval** (Chen et al., 2021) and report the 0-shot pass@1 score. We keep all the demonstrations the same when evaluating different models.

## 5 Results

In this section, we present the results and comparisons from extensive benchmark evaluations. We begin with the synthetic Needle in a Haystack (NIAH) test and then focus on both long-context and standard benchmarks.

### 5.1 Needle in a Haystack

We evaluate the long-context retrieval capabilities of our models using the Needle in a Haystack (NIAH) passkey retrieval test as defined in Moshkammari and Jaggi (2023). This synthetic task serves as a threshold evaluation for assessing whether a language model can maintain awareness of critical information distributed throughout an extremely long input. In this task, the model is challenged to locate a simple passkey—such as a six-digit random number—embedded within a lengthy, nonsensical text sequence.

To quantify retrieval accuracy, we evaluate 40 different input sequence lengths. For each length, the passkey is randomly embedded at 10 uniformly distributed document depths. The results are shown in Figure 2. For our models, we evaluate input lengths up to 1M, 2M, and 4M tokens, while for baseline models, we only evaluate up to 1M tokens. As illustrated in Figures 2a to 2c, among the baseline models, only Llama-3-8B-Instruct-Gradient-1048k passes the NIAH test, while Llama-3.1-8B-Instruct and Llama-3-8B-ProLong-512k-Instruct exhibit errors even within their claimed context lengths. In contrast, as shown in Figures 2d to 2f, our UltraLong models achieve 100% accuracy across all input lengths and depths, demonstrating robust long-context retrieval capability.

### 5.2 Long context evaluation

We present the evaluation results on RULER, LV-Eval, and InfiniteBench in Table 1. Bolded numbers indicate performance that exceeds all baseline models. Overall, our three models consistently

achieve the highest scores in most cases. On the RULER benchmark, UltraLong models obtain the highest average scores for input lengths up to 512K and 1M tokens. For LV-Eval, our models yield the highest average F1 scores within both 128K and 256K token lengths. Additionally, we achieve the best performance on InfiniteBench.

These results confirm that our training recipe effectively extends the context window of language models to ultra-long inputs while maintaining similar performance on the original input length. Among the baseline models, Llama-3.1 is designed for a 128K input length and its performance degrades significantly when the input length exceeds 128K tokens. ProLong, built for a 512K context, performs worse than our models at 512K even though it is trained on substantially more tokens (41B tokens versus 1B tokens in our case). Gradient, the longest-context model among the baselines with support for a 1M input length, exhibits poor performance on LV-Eval and InfiniteBench, suggesting that its design may be overly tuned to synthetic tasks and compromises its effectiveness on real-world tasks. In contrast, our models consistently achieve higher scores across both synthetic (RULER) and hybrid (LV-Eval and InfiniteBench) benchmarks, underscoring the effectiveness and scalability of our approach.

### 5.3 Standard capability evaluation

We further evaluate our models on standard benchmarks across general, math, and code domains to ensure that extending the context length does not compromise short-context task performance. As shown in Table 2, our models achieve performance that is comparable to or higher than that of the base model, Llama-3.1-8B-Instruct, achieving average scores of 62.47, 61.06, and 60.95, respectively, compared to 61.45 for Llama-3.1-8B-Instruct. Notably, our models demonstrate clear improvements on the MMLU and MATH benchmarks, while their performance on other benchmarks such as GSM-8K and HumanEval remains highly competitive.

In contrast, the baseline long-context models, Gradient and ProLong, experience considerable performance degradation on these standard tasks, with average scores of only 37.36 and 40.81, respectively. These results indicate that while our approach effectively extends the context window, it also preserves—and in some cases enhances—the model’s general task capabilities. The significant drop in performance for Llama-3-8B-Instruct-

Model	RULER			LV-Eval		InfiniteBench
	< 128K	< 512K	< 1M	< 128K	< 256K	
Llama-3.1-8B-Instruct (128K)	88.3	68.2	61.3	26.55	23.32	24.66
Llama-3-8B-Instruct-Gradient-1048k	82.4	79.8	77.8	14.14	13.48	28.60
Llama-3-8B-ProLong-512k-Instruct	<b>88.9</b>	79.6	71.2	25.59	24.65	23.54
UltraLong-8B-1M-Instruct	86.6	<b>81.6</b>	<b>79.1</b>	28.07	27.02	32.14
UltraLong-8B-2M-Instruct	85.0	80.2	78.2	<b>30.03</b>	<b>28.90</b>	<b>32.49</b>
UltraLong-8B-4M-Instruct	84.2	80.0	78.0	29.09	28.13	30.38

Table 1: Long context evaluation results on the RULER, LV-Eval, and InfiniteBench benchmarks. Average scores are reported within multiple input lengths—128K, 512K, and 1M for RULER, and 128K and 256K for LV-Eval—while a single aggregate score is provided for InfiniteBench. Bolded numbers indicate the highest performance among the models, demonstrating that our models consistently outperform the baselines.

Model	MMLU	MMLU-Pro	MATH	GSM-8K	HumanEval	Avg
Llama-3.1-8B-Instruct (128K)	64.83	<b>44.33</b>	47.22	<b>81.34</b>	<b>69.51</b>	61.45
Llama-3-8B-Instruct-Gradient-1048k	48.33	34.15	15.12	53.22	35.97	37.36
Llama-3-8B-ProLong-512k-Instruct	65.21	40.23	17.16	71.11	10.36	40.81
UltraLong-8B-1M-Instruct	66.99	42.44	<b>55.10</b>	79.53	68.29	<b>62.47</b>
UltraLong-8B-2M-Instruct	<b>67.31</b>	40.55	51.36	79.00	67.07	61.06
UltraLong-8B-4M-Instruct	65.14	43.28	50.92	77.71	67.68	60.95

Table 2: Evaluation results on standard benchmarks including MMLU, MMLU-Pro, MATH, GSM-8K, and HumanEval, along with the average score across these tasks. Our models demonstrate comparable or superior performance relative to the base model (Llama-3.1-8B-Instruct), while the other long-context models exhibit significant degradation on standard tasks. Bolded numbers indicate the highest performance among the models.

Gradient-1048k and Llama-3-8B-ProLong-512k-Instruct suggests that their methods for long-context training may come at the expense of general task performance.

Overall, our findings demonstrate that our training recipe successfully extends the context length of language models while maintaining strong performance on standard benchmarks.

## 6 Ablation Studies

**Special document separator helps efficient context extension.** To assess the impact of using special characters as document separators during long-context continued pretraining, we conduct an ablation study in which we remove all document separators from our pretraining corpus while keeping all other settings unchanged. The model is trained on the same 1B-token corpus, and the results are shown in Table 3. We find that removing the special document separator leads to a consistent drop in performance across all benchmarks. For example, on the RULER benchmark within 1M token input length, the model without the separator scores 79.15, compared to 80.17 when the

separator is used. Similar trends are observed in LV-Eval and InfiniteBench, where the absence of the special document separator results in performance degradation. These findings indicate that employing special characters as document separators—especially when enabling cross-document attention—enhances training efficiency, achieving higher performance at the same training cost.

**YaRN-based scaling helps scalable context extension.** We also investigate the impact of scaling strategies on context extension by comparing our YaRN-based scaling with the NTK-aware scaling approach (Xu et al., 2024b; Gao et al., 2024; Pekelis et al., 2024). In this experiment, we replace YaRN-based scaling with NTK-aware scaling using  $\theta = 3, 580, 165, 449$ , consistent with the configuration in Gradient. The model is again trained on the same 1B-token corpus, and the results are presented in Table 3. While NTK-aware scaling produces a slightly higher RULER score within 128K tokens (86.91 versus 85.63), it suffers a significant performance drop at extended lengths. For instance, on the RULER benchmark, the NTK-

Model	RULER			LV-Eval		InfiniteBench
	< 128K	< 512K	< 1M	< 128K	< 256K	
UltraLong-8B-1M	85.63	<b>82.28</b>	<b>80.17</b>	<b>27.60</b>	<b>26.40</b>	<b>26.25</b>
w/o special separator	85.47	81.63	79.15	26.06	24.85	22.75
w/ NTK-aware scaling	<b>86.91</b>	80.27	76.62	22.34	21.24	20.18

Table 3: Ablation study results on our continued pretraining strategy for long-context extension. We compare our configuration with two ablated variants: one without the special document separator and one using NTK-aware scaling instead of the YaRN-based scaling. Performance is evaluated on the RULER benchmark for input lengths below 128K, 512K, and 1M tokens, on LV-Eval for input lengths below 128K and 256K tokens, and on InfiniteBench. Bolded scores indicate the best performance among the variants. These results demonstrate the importance of both the special document separator and YaRN-based scaling in achieving robust long-context performance.

Model	RULER			LV-Eval		InfiniteBench
	< 128K	< 512K	< 1M	< 128K	< 256K	
UltraLong-8B-512k-1M	84.22	79.83	77.52	25.15	23.97	<b>26.81</b>
UltraLong-8B-1M	<b>85.63</b>	<b>82.28</b>	<b>80.17</b>	<b>27.60</b>	<b>26.40</b>	26.25
UltraLong-8B-1M-2M	84.60	80.12	78.18	25.68	24.72	24.91
UltraLong-8B-2M	<b>86.30</b>	<b>82.75</b>	<b>80.8</b>	<b>26.00</b>	<b>24.86</b>	<b>25.22</b>

Table 4: Comparison of multi-step and one-step continued pretraining strategies for context extension. Results are presented on the RULER benchmark (for inputs below 128K, 512K, and 1M tokens), LV-Eval (for inputs below 128K and 256K tokens), and InfiniteBench. Bolded scores indicate superior performance within each model pair. With the same training cost, we find that the one-step approach consistently outperforms the multi-step variants.

aware scaling variant achieves 76.62 within 1M tokens, compared to 80.17 with YaRN-based scaling. Similar declines are observed in LV-Eval and InfiniteBench. These results confirm that YaRN-based scaling offers a more robust and scalable method for extending the context window, enabling the model to maintain high performance even as the input length increases.

**One-step continued pretraining is more effective for context extension.** We compare our one-step continued pretraining strategy against a multi-step approach (Gao et al., 2024; Pekelis et al., 2024) for extending the context window. Following the setting in Gao et al. (2024), we split the context extension into two continued pretraining stages. For the 1M model, we first extend the context to 512K tokens by training on 0.5B tokens, and then extend it to 1M tokens with an additional 0.5B tokens—resulting in a total of 1B tokens. We then compare this multi-step model with our one-step model that is directly extended to 1M tokens using 1B tokens of training data. Similarly, for the 2M model, we first extend to 1M and then to 2M, each stage using 0.5B tokens, and compare it with

our one-step 2M model trained on 1B tokens. The results are shown in Table 4. We find that our one-step continued pretraining approach consistently outperforms the multi-step approach across all benchmarks. For example, our one-step 1M model achieves average RULER scores of 85.63, 82.28, and 80.17 within input lengths of 128K, 512K, and 1M, respectively, compared to 84.22, 79.83, and 77.52 for 1M model extended with 2 steps. Similarly, our one-step 2M model outperforms 2M model trained with 2 steps on both RULER and LV-Eval, with higher scores observed on InfiniteBench as well. These results indicate that a one-step extension strategy is more efficient and effective than a multi-step training process.

## 7 Limitations and Future Work

Our current work focuses on SFT on instruction datasets during the instruction tuning stage, and does not explore reinforcement learning or preference optimization, which we leave for future work. Additionally, while we present an effective recipe for extending long-context capabilities, our work does not address safety alignment. As a result, potential risks such as the generation of harmful

or misleading information and privacy concerns remain unmitigated. Future research will aim to integrate safety alignment mechanisms and explore advanced tuning strategies to further enhance both performance and trustworthiness.

## 8 Conclusion

In this work, we introduce an efficient and systematic training recipe for ultra-long context language models, extending context windows to 1M, 2M, and 4M tokens while maintaining competitive performance on standard benchmarks. Our approach combines efficient continued pretraining with instruction tuning to enhance both long-context understanding and instruction-following capabilities. Extensive experiments and ablation studies on our UltraLong-8B model series validate our design choices, including the use of special document separators, YaRN-based scaling, and an efficient one-step training strategy. We believe our framework sets a new standard for scalable long-context modeling and paves the way for future research aimed at improving long-context performance in practical applications.

## Acknowledgments

Chejian Xu and Bo Li are partially supported by the National Science Foundation under grant No. 1910100, No. 2046726, NSF AI Institute ACTION No. IIS-2229876, DARPA TIAMAT No. 80321, the National Aeronautics and Space Administration (NASA) under grant No. 80NSSC20M0229, ARL Grant W911NF-23-2-0137, Alfred P. Sloan Fellowship, the research grant from AI Safety Fund, Virtue AI, and Schmidt Science.

## References

- 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*.
- Anthropic. 2024. [Claude 3.5 sonnet](#).
- Alisson Azzolini, Hannah Brandon, Prithvijit Chattopadhyay, Huayu Chen, Jinju Chu, Yin Cui, Jenna Diamond, Yifan Ding, Francesco Ferroni, Rama Govindaraju, and 1 others. 2025. Cosmos-reason1: From physical common sense to embodied reasoning. *arXiv preprint arXiv:2503.15558*.
- bloc97. 2023. [Ntk-aware scaled rope allows llama models to have extended \(8k+\) context size without any fine-tuning and minimal perplexity degradation](#).
- Guanzheng Chen, Xin Li, Zaiqiao Meng, Shangsong Liang, and Lidong Bing. 2023a. Clx: Continuous length extrapolation for large language models. *arXiv preprint arXiv:2310.16450*.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.
- 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*.
- 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*.
- Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, and 1 others. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%\* chatgpt quality. See <https://vicuna.lmsys.org> (accessed 14 April 2023), 2(3):6.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Wenliang Dai, Nayeon Lee, Boxin Wang, Zhuolin Yang, Zihan Liu, Jon Barker, Tuomas Rintamaki, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. NVLM: Open frontier-class multimodal LLMs. *arXiv preprint arXiv:2409.11402*.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- emozilla. 2023. [Dynamically scaled rope further increases performance of long context llama with zero fine-tuning](#).
- Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hananeh Hajishirzi, Yoon Kim, and Hao Peng. 2024. Data engineering for scaling language models to 128k context. *arXiv preprint arXiv:2402.10171*.
- Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. 2024. How to train long-context language models (effectively). *arXiv preprint arXiv:2410.02660*.
- Gemini, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, and 1 others.

2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.
- Glaive-AI. 2023. [Glaivecodeassistant](#).
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shitong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-R1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. 2024. [LM-infinite: Zero-shot extreme length generalization for large language models](#). pages 3991–4008, Mexico City, Mexico.
- 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*.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.
- Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekeshe, 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*.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Dongseong Hwang, Weiran Wang, Zhuoyuan Huo, Khe Chai Sim, and Pedro Moreno Mengibar. 2024. Transformerfam: Feedback attention is working memory. *arXiv preprint arXiv:2404.09173*.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.
- Wing Lian, Guan Wang, Bleyds Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknum". 2023. [Slimorca: An open dataset of gpt-4 augmented flan reasoning traces, with verification](#).
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024a. Deepseek-V3 technical report. *arXiv preprint arXiv:2412.19437*.
- Zihan Liu, Yang Chen, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024b. AceMath: Advancing frontier math reasoning with post-training and reward modeling. *arXiv preprint arXiv:2412.15084*.
- Yi Lu, Jing Nathan Yan, Songlin Yang, Justin T Chiu, Siyu Ren, Fei Yuan, Wenting Zhao, Zhiyong Wu, and Alexander M Rush. 2024. A controlled study on long context extension and generalization in llms. *arXiv preprint arXiv:2409.12181*.
- Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2023. Wizardcoder: Empowering code large language models with evolve-instruct. *arXiv preprint arXiv:2306.08568*.
- Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. 2024. Orca-math: Unlocking the potential of slms in grade school math. *arXiv preprint arXiv:2402.14830*.
- Amirkeivan Mohtashami and Martin Jaggi. 2023. Landmark attention: Random-access infinite context length for transformers. *arXiv preprint arXiv:2305.16300*.
- Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. [Orca: Progressive learning from complex explanation traces of gpt-4](#). *Preprint, arXiv:2306.02707*.
- OpenAI. 2024. Gpt-4o mini: advancing cost-efficient intelligence. URL: <https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/>.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Leonid Pekelis, Michael Feil, Forrest Moret, Mark Huang, and Tiffany Peng. 2024. [Llama 3 gradient: A series of long context models](#).
- Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023a. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.
- Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023b. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*.
- Liliang Ren, Yang Liu, Yadong Lu, Yelong Shen, Chen Liang, and Weizhu Chen. 2024. Samba: Simple hybrid state space models for efficient unlimited context language modeling. *arXiv preprint arXiv:2406.07522*.

- 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*.
- Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.
- Teknum. 2023. [Gpteacher-general-instruct](#).
- The-Vicuna-Team. 2023. [ShareGPT-Vicuna](#).
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, and 1 others. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. 2024. Magicoder: Empowering code generation with oss-instruct. In *Forty-first International Conference on Machine Learning*.
- Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024a. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.
- Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2023. Retrieval meets long context large language models. *arXiv preprint arXiv:2310.03025*.
- Peng Xu, Wei Ping, Xianchao Wu, Chejian Xu, Zihan Liu, Mohammad Shoeybi, and Bryan Catanzaro. 2024b. ChatQA 2: Bridging the gap to proprietary llms in long context and RAG capabilities. *arXiv preprint arXiv:2407.14482*.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.
- An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, and 1 others. 2025. Qwen2. 5-1m technical report. *arXiv preprint arXiv:2501.15383*.
- Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.
- Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, and 1 others. 2024a. Advancing llm reasoning generalists with preference trees. *arXiv preprint arXiv:2404.02078*.
- Tao Yuan, Xuefei Ning, Dong Zhou, Zhijie Yang, Shiyao Li, Minghui Zhuang, Zheyue Tan, Zhuyu Yao, Dahua Lin, Boxun Li, and 1 others. 2024b. Lv-eval: A balanced long-context benchmark with 5 length levels up to 256k. *arXiv preprint arXiv:2402.05136*.
- Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhua Chen. 2023. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*.
- Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024. [∞Bench: Extending long context evaluation beyond 100K tokens](#). pages 15262–15277, Bangkok, Thailand.
- Liang Zhao, Tianwen Wei, Liang Zeng, Cheng Cheng, Liu Yang, Peng Cheng, Lijie Wang, Chenxia Li, Xuejie Wu, Bo Zhu, and 1 others. 2024. Longskywork: A training recipe for efficiently extending context length in large language models. *arXiv preprint arXiv:2406.00605*.

<b>Continued Pretraining</b>	
<b>Data</b>	Per-source upsampled pretraining corpus. Documents are separated using “<s>”.
<b>Length</b>	1M, 2M, and 4M
<b>Steps</b>	1B tokens
<b>Model</b>	Initialization: Llama-3.1-8B-Instruct (RoPE base freq. $5 \times 10^5$ , RoPE scaling factor $s = 8$ ) RoPE scaling: $s = 128, 256, \text{ and } 512$ Attention: Full attention without cross-document attention masking
<b>Optim.</b>	Adam ( $\beta_1 = 0.9, \beta_2 = 0.95$ ) LR: $3e-05$
<b>Instruction Tuning</b>	
<b>Data</b>	General: ShareGPT, SlimOrca, EvolInstruct, GPTeacher, AlpacaGPT4, and UltraInteract Mathematics: OrcaMathWordProblems, MathInstruct, and MetaMath Code: Magicoder, WizardCoder, and GlaiiveCodeAssistant
<b>Steps</b>	1B tokens
<b>Optim.</b>	Adam ( $\beta_1 = 0.9, \beta_2 = 0.95$ ) LR: $5e-06$

Table 5: Overview of our training recipe for UltraLong-8B-Instruct models.

## A UltraLong Recipe

Table 5 provides a detailed overview of our training recipe, which is divided into two stages: continued pretraining and instruction tuning. In the continued pretraining phase, we utilize per-source upsampled pretraining data as Fu et al. (2024) to extend the context window to 1M, 2M, and 4M tokens over 1B tokens of training. The model is initialized from Llama-3.1-8B-Instruct with RoPE-based positional embeddings (base frequency  $5 \times 10^5$  and initial scaling factor  $s = 8$ ), and extended using scaling factors of  $s = 128, 256, \text{ and } 512$ , with full attention applied without cross-document masking. In the instruction tuning stage, we fine-tune the model using a diverse dataset covering general, mathematics, and code domains, again training on 1B tokens with an Adam optimizer at a learning rate of  $5 \times 10^{-6}$ .