LONGEMBED: EXTENDING EMBEDDING MODELS FOR LONG CONTEXT RETRIEVAL

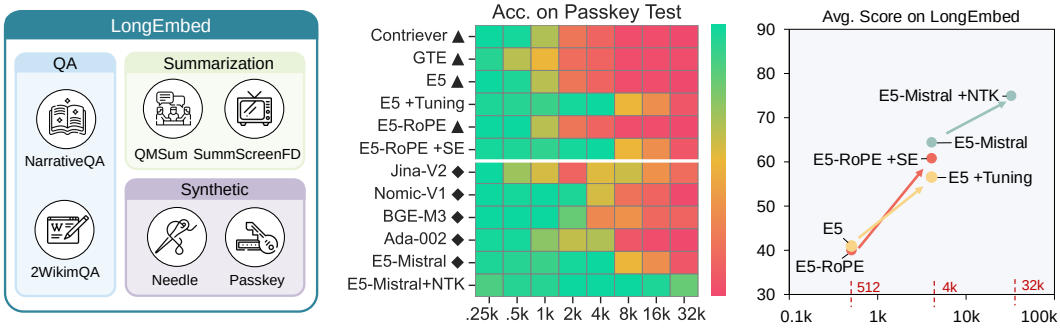
Dawei Zhu [*η] **Liang Wang** [π] **Nan Yang** [π] **Yifan Song** [η] **Wenhao Wu** [η]
Furu Wei [π] **Sujian Li** [η]

[η] Peking University [π] Microsoft Corporation

https://github.com/dwzhu-pku/LongEmbed

ABSTRACT

Embedding models play a pivot role in modern NLP applications such as IR and RAG. While the context limit of LLMs has been pushed beyond 1 million tokens, embedding models are still confined to a narrow context window not exceeding 8k tokens, refrained from application scenarios requiring long inputs such as legal contracts. This paper explores context window extension of existing embedding models, pushing the limit to 32k without requiring additional training. First, we examine the performance of current embedding models for long context retrieval on our newly constructed LONGEMBED benchmark. LONGEMBED comprises two synthetic tasks and four carefully chosen real-world tasks, featuring documents of varying length and dispersed target information. Benchmarking results underscore huge room for improvement in these models. Based on this, comprehensive experiments show that training-free context window extension strategies like position interpolation can effectively extend the context window of existing embedding models by several folds, regardless of their original context being 512 or beyond 4k. Furthermore, for models employing absolute position encoding (APE), we show the possibility of further fine-tuning to harvest notable performance gains while strictly preserving original behavior for short inputs. For models using rotary position embedding (RoPE), significant enhancements are observed when employing RoPE-specific methods, such as NTK and SelfExtend, indicating RoPE's superiority over APE for context window extension. To facilitate future research, we release $E5_{Base}$-4k and $E5\text{-}RoPE_{Base}$, along with the LONGEMBED benchmark.



(a) (b) (c)

Figure 1: **(a)** Overview of the LONGEMBED benchmark. **(b)** Performance of current embedding models on passkey retrieval, with evaluation length ranging from 256 to 32,768.[1] ▲ / ♦ denotes embedding models with 512 / $\geq$ 4k context. The greener ■ a cell is, the higher retrieval accuracy this model achieves on the corresponding evaluation length. **(c)** Effects of context window extension methods on E5, E5-RoPE, E5-Mistral, measured by improvements of Avg. Scores on LONGEMBED. SE / NTK is short for SelfExtend / NTK-Aware Interpolation.

[*] Work done during Dawei's internship at MSR Asia. Prof. Sujian Li is the corresponding author.

[1] For simplicity, we report results from the *base* versions of the included models by default.

1 INTRODUCTION

Text embeddings are vector representations of natural language that encode its semantic information. They play a pivotal role in various natural language processing (NLP) tasks, including information retrieval (IR) and retrieval-augmented generation (RAG). However, embedding models for producing these vector representations still operates within a very narrow context window, typically 512 input tokens (Wang et al., 2022; Xiao et al., 2023; Ni et al., 2022), This narrow context window has greatly hindered their application in scenarios requiring long inputs, such as long wikipedias and meeting scripts (Saad-Falcon et al., 2024).

Previous efforts that train a long context embedding model *from scratch* suffer significant computational overhead, due to the combined demand for large batch sizes and long sequences. For example, Chen et al. (2024) utilized 96 A100 GPUs to train BGE-M3 which supports 8k context. Meanwhile, there have been many successes in extending context window of *existing* LLMs in a plug-and-play way or via efficient fine-tuning, pushing their context from 4k to 128k (Xiong et al., 2023) and even 2 million tokens (Ding et al., 2024). Motivated by this, instead of training long context embedding models from scratch, this paper explores context window extension of *existing* embedding models.

First, we examine the capability of existing embedding models in processing long context. Retrieval is selected as the proxy task, as it closely mirrors real-world application scenarios. While there have been some retrieval benchmarks such as BEIR (Thakur et al., 2021) and LoCo (Saad-Falcon et al., 2024), we identify two major limitations with these existing benchmarks: 1) limited document length, 2) biased distribution of target information. To overcome this, we introduce the LONGEMBED benchmark that integrates two synthetic tasks to enable flexible control over document length, and four real tasks featuring dispersed target information. Results on LONGEMBED indicates huge room for improvement in current embedding models.

Based on this, we explore plug-and-play strategies to extend embedding models, including parallel context windows, reorganizing position ids, and position interpolation. Comprehensive experiments show that these strategies can effectively extend the context window of existing embedding models by several folds, regardless of their original context being 512 or beyond 4k. Furthermore, for models employing absolute position encoding (APE), we show the possibility of harvesting further improvements via fine-tuning while strictly preserving original behavior within the short context. In this way, we have extended E5_{Base} (Wang et al., 2022) from 512 to 4k (See Figure 1c).

For models utilizing RoPE (Su et al., 2021), substantial enhancements on LONGEMBED are observed when employing methods that fully leverage RoPE's advantages, such as NTK (Peng & Quesnelle, 2023) and SelfExtend (Jin et al., 2024). As illustrated in Figure 1b and 1c, leveraging NTK extends the context window of E5-Mistral to 32k, achieving close-to-perfect accuracy on passkey retrieval and state-of-the-art performance on LONGEMBED. Further, for fair comparison of APE / RoPE-based embedding models, we pre-train E5-RoPE following the training procedure and data of E5. Thorough comparison of E5 and E5-RoPE reveals the superiority of RoPE-based embedding models in context window extension.

To facilitate future research in long context embedding models, we release E5_{Base}-4k, $\text{E5-RoPE}_{\text{Base}}$, and the LongEmbed benchmarks. E5_{Base}-4k is further fine-tuned on E5_{Base} to support 4k context, while strictly preserving original behavior for inputs not exceeding 512 tokens. $\text{E5-RoPE}_{\text{Base}}$ follows the same training procedure as E5_{Base}, except for the substitution of APE with RoPE. It is released to facilitate comparison between APE & RoPE-Based embedding models. Furthermore, we have integrated LONGEMBED into MTEB (Muennighoff et al., 2023) to make evaluation more convenient.

2 RELATED WORK

Text Embedding Models. Text embeddings are continuous, low-dimensional vector representations of text that encode semantic information, laying the foundation of numerous NLP applications. Early attempts on text embeddings includes latent semantic indexing (Deerwester et al., 1990) and weighted average of word embeddings (Mikolov et al., 2013). Modern embedding models (Wang et al., 2022; Xiao et al., 2023; Neelakantan et al., 2022) exploit supervision from labeled query-document pairs, adopting a multi-stage training paradigm, where they are first pre-trained on large-scale weakly-supervised text pairs using contrastive loss, then fine-tuned on small scale but high-quality datasets.

More recently, Muennighoff et al. (2024) explores the combination of generative and embedding tasks on LLMs, introducing GritLM that harvests improvements in both aspects.

Existing efforts in developing long-context embedding models typically involve first obtaining a long-context backbone model, either by pre-training with long inputs from scratch (Günther et al., 2023; Nussbaum et al., 2024; Chen et al., 2024) or using existing ones (Wang et al., 2023b), followed by training the backbone model to produce embeddings. Instead, this paper endows *existing* embedding models with the ability to handle long context through context window extension.

Context Window Extension for Large Language Models. Due to the high cost of pre-training an LLM from scratch, there have been many efforts towards extending the context window of *existing* LLMs in a plug-and-play manner. We categorize these efforts as follows: 1) *Divide-and-conquer*, which involves segmenting long inputs into short chunks, processing each chunk with the model, and aggregating the results, as demonstrated by PCW (Ratner et al., 2023); 2) *Position reorganization*, which reorganizes position ids to boost length extrapolation, as exemplified by SelfExtend (Jin et al., 2024), DCA (An et al., 2024), and others; 3) *Position interpolation*, which introduces new position embeddings by interpolating existing ones, includes PI (Chen et al., 2023), NTK (Peng & Quesnelle, 2023), YaRN (Peng et al., 2023), and Resonance RoPE (Wang et al., 2024a). Our paper thoroughly investigates these three lines of methods on embedding models. We also acknowledge other efforts for extending the context window, such as prompt & KV compression (Jiang et al., 2023; Ge et al., 2023; Zhang et al., 2024a) and memory-based transformers (Wang et al., 2024b; Xiao et al., 2024). However, the former is not applicable for bidirectional attention, and the latter requires complex mechanisms for accessing encoded content, hence we do not experiment with these two categories.

In addition to their plug-and-play usability, further fine-tuning on top of these methods with long training samples has been proven to yield better performance (Xiong et al., 2023; Fu et al., 2024; Zhang et al., 2024b; Yen et al., 2024). Addressing the overhead of training on long inputs and the scarcity of extremely long training data, a line of research investigates simulating long inputs within short context, including Randomized Positions (Ruoss et al., 2023), Positional Skip-wise (PoSE) training (Zhu et al., 2023). This paper also leverage these efforts to synthesize long training samples from the original training data, facilitating further fine-tuning on top of plug-and-play methods.

3 The LongEmbed benchmark

In this section, we first identify two limitations of existing retrieval benchmarks for evaluating long-context capabilities (Section 3.1). Then, we introduce the retrieval tasks adopted in our LongEmbed, including both synthetic ones (Section 3.2) and real ones (Section 3.3).

3.1 Examination of Existing Retrieval Benchmarks

There are mainly two desiderata for curating a benchmark for long context retrieval. First, the candidate documents should be long enough. Second, the target information to answer user query should be as uniformly distributed across the document as possible. This prevents embedding models from solely focusing on specific parts, such as the beginning (Coelho et al., 2024), to achieve unreasonably high scores. Based on these criteria, we evaluate existing benchmarks for text retrieval as follows:

BEIR Benchmark (Thakur et al., 2021) is a collection of 18 information retrieval datasets, ranging across ad-hoc web search, question answering, fact verification and duplicate question retrieval, etc. However, documents in this benchmark contains fewer than 300 words on average (See Table 5 in Appendix), making it unsuitable for measuring long context retrieval that usually involves documents of thousands or tens of thousands of words.

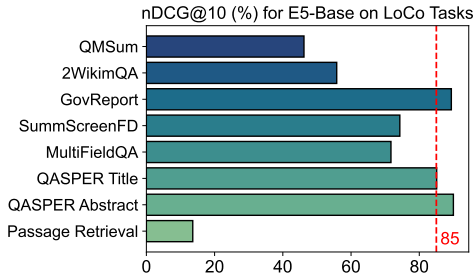


Figure 2: Results of $E5_{Base}$ on 8 LoCo tasks.

LoCo Benchmark (Saad-Falcon et al., 2024) consists 12 retrieval tasks that requires long context reasoning, spanning diverse domains such as law, science, finance, etc. However, we show that it still

suffers from biased distribution of key information. Figure 2 presents results of $E5_{Base}$ on 8 LoCo tasks that are publicly available. With only 512 context length, $E5_{Base}$ achieves >85% nDCG scores on 3 out of 8 retrieval tasks. This severely biased distribution of target information undermines its ability to reflect model performance as context length increases.

3.2 Synthetic Tasks in LongEmbed

First, we tailor the passkey retrieval and needle-in-a-haystack retrieval task designed for LLMs to measure context length of embedding models as follows:

Personalized Passkey Retrieval. Passkey retrieval (Mohtashami & Jaggi, 2023) requires LLMs to recover a random passkey hidden within a long document comprising garbage information. For embedding models, we adopt the personalized passkey retrieval task proposed by Wang et al. (2023b), where each document contains a unique person name and his/her passkey at random position. The goal is to retrieve the document containing the given person's passkey from all candidates documents.

Needle-in-a-haystack Retrieval. While passkey retrieval surrounds key information with garbage sentences, needle-in-a-haystack retrieval (Kamradt, 2023) randomly inserts key information into an arbitrary position of a long essay, making the task more challenging. To tailor this task for embedding models, we instruct GPT-4 to generate 100 facts covering a variety of domains including physics, history, geometry, art, etc, and 100 *queries* correspondingly. The facts are treated as *needles* and randomly inserted into the PaulGrahamEssay to form 100 candidate documents. Our task is to correctly retrieve the document that contains corresponding needle given the query.

Passkey Test Examples:
Query: What is the pass key for Sky Morrow?
Doc1: <prefix> Sky Morrow's passkey is 123. Remember it. 123 is the passkey for Sky Morrow. <suffix>
Doc2: <prefix> Cesar McLean's passkey is 456. Remember it. 456 is the passkey for Cesar McLean. <suffix>
...

Needle Test Examples:
Query: Who discovered the law of gravity?
Doc1: <prefix> The law of gravity was discovered by Sir Issac Newton. <suffix>
Doc2: <prefix> The best thing to do in San Francisco is eat a sandwich and sit in Dolores Park on a sunny day. <suffix>
...

Figure 3: Example for the passkey and needle test.

The advantage of synthetic data is that we can flexibly control context length and distribution of target information. For both tasks, we evaluate a broad context range of $\{0.25, 0.5, 1, 2, 4, 8, 16, 32\} \times 1024$ tokens [2]. For each context length, we include 50 test samples, each comprising 1 query and 100 candidate documents. [3] In this way, we can measure the effective context size of embedding models for up to 32k tokens. Examples for both synthetic tasks are presented in Figure 3. For the passkey test, the <*prefix / suffix*> are repeats of *"The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again."* For the needle test, the <*prefix*> and <*suffix*> form a long essay.

3.3 Real Tasks in LongEmbed

While synthetic tasks offer flexibility in manipulating context length and distributing key information, they still differ from real-world scenarios. To conduct a comprehensive evaluation, we have tailored following long-form QA and summarization tasks for long context retrieval. Note that for QA and summarization datasets, we use the questions and summaries as queries, respectively.

NarrativeQA (Kočiský et al., 2018) is a QA dataset comprising long stories averaging 50,474 words and corresponding questions about specific content such as characters, events. As these details are dispersed throughout the story, models must process the entire long context to get correct answers.

2WikiMultihopQA (Ho et al., 2020) is a multi-hop QA dataset featuring questions with up to 5 hops, synthesized through manually designed templates to prevent shortcut solutions. This necessitates the ability to process and reason over long context, ensuring that answers cannot be obtained by merely focusing on a short span within the document.

[2] Since token numbers vary w.r.t. tokenizers, we use a rough estimation that 1 token = 0.75 word, and constraint the word numbers to not exceed $\{0.25, 0.5, 1, 2, 4, 8, 16, 32\} \times 1024 \times 0.75$.

[3] The original version of personalized passkey retrieval uses different candidate documents for each query, resulting in 50 queries and 5,000 documents to encode for each context length. To speed up evaluation, we share the candidate documents for different queries within each context length.

Table 1: Overview of the LONGEMBED benchmark. Average word number is rounded to the nearest integer. † For needle and passkey test, we have 8 groups of queries and candidate documents, with the documents averaging $\{0.25, 0.5, 1, 2, 4, 8, 16, 32\} \times 0.75k$ words, respectively.

Dataset	Domain	# Queries	# Docs	Avg. Query Words	Avg. Doc Words
		Real Tasks			
NarrativeQA	Literature, Film	10,449	355	9	50,474
QMSum	Meeting	1,527	197	71	10,058
2WikimQA	Wikipedia	300	300	12	6,132
SummScreenFD	ScreenWriting	336	336	102	5,582
		Synthetic Tasks			
Passkey	Synthetic	400	800	11	†
Needle	Synthetic	400	800	7	†

QMSum (Zhong et al., 2021) is a query-based meeting summarization dataset that requires selecting and summarizing relevant segments of meetings in response to queries. Due to the involvement of multiple participants and topics in the meeting, summarization regarding specific queries naturally requires aggregating information dispersed throughout the entire text.

SummScreenFD (Chen et al., 2022) is a screenplay summarization dataset comprising pairs of TV series transcripts and human-written summaries. Similar to QMSum, its plot details are scattered throughout the transcript and must be integrated to form succinct descriptions in the summary.

Table 1 presents the overall statistics of LONGEMBED. Considering the computational complexity that increases quadratically with input length, we intentionally restrict the number of candidate documents in each task to to not exceed 10^3. In this way, we can efficiently evaluate the basic long context capabilities of embedding models. For further elaboration on the source and examples for each dataset, please refer to Appendix C.

4 METHODOLOGY

4.1 ABSOLUTE POSITION EMBEDDING (APE) & ROTARY POSITION EMBEDDING (ROPE)

Absolute Position Embedding (APE) stands as the predominant positional encoding strategy for embedding models, as majority of them follows the BERT architecture (Devlin et al., 2019). APE-based models first embed absolute position ids into position vectors and add token embeddings to their corresponding position vectors, before feeding them to a stack of transformer layers.

Rotary Position Embedding (RoPE) is the most pervasive position embedding strategy in the era of LLMs, including LLaMA (Touvron et al., 2023), Gemma (Team et al., 2024), QWen (Bai et al., 2023a), etc. It encodes position information of tokens with a rotation matrix that naturally incorporates explicit relative position dependency. To elucidate, given a hidden vector $\boldsymbol{h} = [h_0, h_1, ..., h_{d-1}]$ of dimension d, and a position index m, RoPE operates as follows:

$$f(\boldsymbol{h}, m) = [(h_0 + \mathrm{i}h_1)e^{\mathrm{i}m\theta_0}, (h_2 + \mathrm{i}h_3)e^{\mathrm{i}m\theta_1}, ..., (h_{d-2} + \mathrm{i}h_{d-1})e^{\mathrm{i}m\theta_{d/2-1}}] \tag{1}$$

where $\theta_j = 10000^{-2j/d}, j \in \{0, 1, ..., d/2-1\}$, $\mathrm{i} = \sqrt{-1}$ is the imaginary unit. Unlike APE that is directly applied to the input vector $\boldsymbol{x}$, RoPE is employed on the query and key vectors at each layer. The attention score $a(\boldsymbol{q}, \boldsymbol{k})$ between a query $\boldsymbol{q}$ at position m and a key $\boldsymbol{k}$ at position n is defined as:

$$\begin{aligned} a(\boldsymbol{q}, \boldsymbol{k}) &= \mathrm{Re}\langle f(\boldsymbol{q}, m), f(\boldsymbol{k}, n)\rangle = \mathrm{Re}\left[\sum_{j=0}^{d/2-1} (q_{2j} + \mathrm{i}q_{2j+1})(k_2 j - \mathrm{i}k_{2j+1})e^{\mathrm{i}(m-n)\theta_j}\right] \\ &:= g(\boldsymbol{q}, \boldsymbol{k}, (m-n)\boldsymbol{\theta}) \end{aligned} \tag{2}$$

where g(·) is an abstract mapping function exclusively dependent on $\boldsymbol{q}, \boldsymbol{k}$ and $(m-n)\boldsymbol{\theta}$.

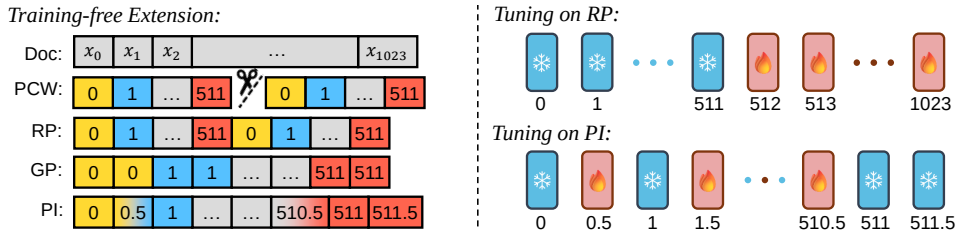


Figure 4: (Left) Arrangement of pids for extending APE-based models from 512 to 1024. (Right) Illustration of learnable (🔥) and frozen (❄) position vectors when further tuning on RP / PI.

4.2 Context Window Extension for APE-based Models

As delineated in Section 2, training-free context extension strategies applicable to embedding models can be classified into 3 categories: 1) Divide-and-conquer; 2) Position reorganization; 3) Position interpolation. In this section, we introduce methods from each of these categories to assess their applicability to embedding models. Further fine-tuning on top of these methods is also included. Let L_o represent the original context length, $\mathcal{D} = \{x_1, x_2, ..., x_{L_t}\}$ denote a long document of target context length L_t, and $s = \lceil L_t/L_o \rceil$ indicate the context scaling factor. The context extension methods we investigated are described below:

Parallel Context Windows (PCW). To process a long document with a short-context model, PCW divides the long document into multiple short chunks, processes each chunk in parallel, and aggregates their results (Ratner et al., 2023; Yen et al., 2024). In practice, we first segment $\mathcal{D}$ into chunks of L_o tokens, then average over each chunk's embeddings to get the embedding of $\mathcal{D}$. For simplicity, we set the overlap between adjacent chunks to 0, except for the last chunk, which conditionally overlaps with the preceding chunk to ensure it contains L_o tokens.

Grouped Positions (GP) & Recurrent Positions (RP). Dividing inputs into chunks and processing them separately sacrifices their interaction in between. By contrast, position reorganization accommodates longer context by reusing the original position ids. To be specific, we experiment with two simple strategies: *Grouped Positions* and *Recurrent Positions*. The former groups the original position ids as such: $f_{gp}(pid) \rightarrow \lfloor pid/s \rfloor$, while the latter assigns the position ids recurrently within the range $\{0, 1, ..., L_o - 1\}$, formulated as: $f_{rp}(pid) \rightarrow pid \bmod L_o$.

Linear Position Interpolation (PI). Instead of reusing position ids, Chen et al. (2023) introduces new position embeddings via linear interpolation of existing ones. To apply PI on APE-based models, we map the positions ids as such: $f_{pi}(pid) \rightarrow pid/s$, and assign embeddings for non-integers as linear interpolation of that of neighboring integers. In practice, we first extend the original position embedding matrix $E_o \in \mathbb{R}^{L_o \times d}$ into $E_t \in \mathbb{R}^{L_t \times d}$, where d stands for hidden size. Next, we assign $E_t[i \cdot s] = E_o[i], i \in \{0, 1, ..., L_o - 1\}$. For non-integer position id j between i and $i + 1$, we determine their embeddings as follows: $E_t[s \cdot j] = ((i + 1 - j)E_t[i \cdot s] + (j - i)E_t[(i + 1) \cdot s])$.

Further Tuning. Except for PCW, which divides long texts into smaller blocks and processes separately, GP, RP, and PI can all be seen as extending the position embedding matrix. Since APE-based models assign an independent vector to each position, we can freeze the original model parameters while updating only the newly added position embeddings. In this way, we can strictly maintain model ability within 512 context, while harvesting further performance gains in handling long context as free lunch. Specifically, further fine-tuning on top of RP and PI is explored in this paper, as illustrated in Figure 4 (Right). Since the traditional training data for embedding models are short queries and passages not exceeding 512 tokens, we manipulate position ids to simulate long training samples, as proposed in Zhu et al. (2023). See Appendix B for details of further fine-tuning.

4.3 Context Window Extension for RoPE-based Models

For RoPE-based models, we further explore Self Extend and NTK, which respectively advances over GP and PI, harnessing the inherent advantages of RoPE. Since there is no simple strategy for further training while exactly maintaining original performance like APE, we leave comprehensive exploration of training-based context window extension for RoPE-based models for future work.

Self Extend (SE). Compared with APE, RoPE operates on the query and key vectors at each layer to encode relative positions, offering enhanced flexibility for position reorganization. For each token, instead of assigning grouped relative positions to all other tokens, SelfExtend (Jin et al., 2024) re-introduces normal relative positions within the nearest neighbor window w, achieving improved performance. For example, consider a document of 10 tokens $\{x_0, x_1, ..., x_9\}$ with a neighbor window size $w = 4$ and a group size $g = 2$. The relative positions for x_0 are $\{0, 1, 2, 3, 4, 4, 5, 5, 6, 6\}$. For x_4, the relative positions of the other tokens are $\{-4, -3, -2, -1, 0, 1, 2, 3, 4, 4\}$.

NTK-Aware Interpolation (NTK). Given a scaling factor s, PI proportionally down-scales position index m to m/s. In this way, the attention score $a(\boldsymbol{q}, \boldsymbol{k})$ defined in Equation 2 becomes $g(\boldsymbol{q}, \boldsymbol{k}, (m-n)\boldsymbol{\theta}/s)$. This is also equivalent to reducing the frequencies $\boldsymbol{\theta}$ uniformly, which may prevent the model from learning high-frequency features, as shown by the Neural Tangent Kernel (NTK) theory (Jacot et al., 2018). To remedy this, NTK-Aware interpolation (Peng & Quesnelle, 2023) scales high frequencies less and low frequencies more to spread out the interpolation pressure across multiple dimensions. This is achieved by directly altering the original $\theta_j = 10000^{-2j/d}$ into $\theta'_j = (10000\lambda)^{-2j/d}$, where λ is conventionally chosen to be slightly greater than s.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Benchmarked Models. We evaluate both open-sourced and proprietary models on LONGEMBED, including E5_{Base} (Wang et al., 2022), GTE_{Base} (Li et al., 2023), BGE-Base (Xiao et al., 2023), Contriever (Izacard et al., 2021), GTR-Base (Ni et al., 2022), E5-Mistral (Wang et al., 2023b), Jina-V2 (Günther et al., 2023), Nomic-V1 (Nussbaum et al., 2024), BGE-M3 (Chen et al., 2024), OpenAI-ada-002. For BGE-M3, we utilize dense vectors. M2 (Saad-Falcon et al., 2024) is not included in our evaluation, given its training data partly overlaps with test samples in LONGEMBED.

Candidate Models for Extension. From each of the APE-based and RoPE-based category, we select 2 candidate models for comprehensive study. The former includes E5_{Base} and GTE_{Base}. The latter includes the 4,096-context E5-Mistral, and a newly trained $\text{E5-RoPE}_{\text{Base}}$, which supports 512 context (See Appendix A for its training details and BEIR results). Note that $\text{E5-RoPE}_{\text{Base}}$ employs the same training procedure and training data as E5_{Base}, only with APE substituted with RoPE. This facilitates fair comparison of APE / RoPE-based models in context window extension, as presented in Section 5.4. For implementation details of each context window extension strategies on each model, please refer to Appendix B.

5.2 MAIN RESULTS

Table 2 demonstrates the performance of existing embedding models on our LONGEMBED benchmark. Among the 512-context models, E5_{Base} achieves the highest average score of 41.0 points, closely followed by $\text{E5-RoPE}_{\text{Base}}$ and Contriever. As the supported context length increases beyond 4k, exemplified by E5-Mistral and Jina-V2, a discernible increase in scores is observed. This verifies both the efficacy of these long-context models and the validity of LONGEMBED to assess long-context retrieval. Note that even the best performing model attains only 64.4 pts on average, indicating huge room for improvement in current models.

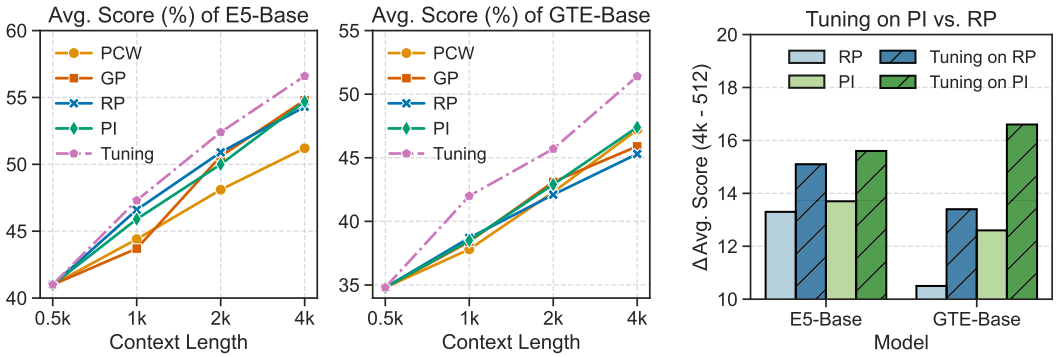
In the last row block of Table 2, we further include the best results achieved by E5_{Base}, $\text{E5-RoPE}_{\text{Base}}$ and E5-Mistral after context window extension. For E5_{Base} and $\text{E5-RoPE}_{\text{Base}}$, we extend their contexts from 512 to 4,096. For E5-Mistral, we extend its context from 4,096 to 32,768. Compared to the original versions, the extended models achieve an average score increase of +15.6 / +20.3 / +10.9 points. This indicates the efficacy of these context extension strategies on embedding models, enabling them to handle inputs of several folds longer. Detailed performance comparison of different extension strategies on APE & RoPE-based embedding models is presented in Section 5.3.

5.3 PERFORMANCE COMPARISON OF CONTEXT EXTENSION METHODS

APE-Based Models. Figure 5a illustrates the impact of various context extension strategies on E5_{Base} and GTE_{Base} across different target context lengths. We observe that plug-and-play methods

Table 2: Results (%) of existing and extended embedding models on LONGEMBED. *NQA*, *SFD*, *2WmQA* is short for *NarrativeQA*, *SummScreenFD*, *2WikiMultihopQA*, respectively. We show that context window extension can effectively improve existing embedding models in handling long context.

Model	Param.	Synthetic (Acc@1)		Real (nDCG@10)				Avg.
		Passkey	Needle	NQA	QMSum	SFD	2WmQA	
512 Context Models								
$E5_{Base}$ (Wang et al., 2022)	110M	38.0	28.5	25.3	23.8	74.7	55.8	**41.0**
E5-$RoPE_{Base}$	110M	38.5	31.5	24.6	23.2	66.6	58.8	40.5
GTE_{Base} (Li et al., 2023)	110M	31.0	24.5	28.6	21.8	55.8	47.3	34.8
BGE-Base (Xiao et al., 2023)	110M	18.0	25.3	25.6	22.4	60.3	51.7	33.9
Contriever (Izacard et al., 2021)	110M	38.5	29.0	26.7	25.5	73.5	47.3	40.1
GTR-Base (Ni et al., 2022)	110M	38.5	26.3	26.5	18.3	63.7	52.2	36.5
$\geq$ 4k Context Models								
E5-Mistral (Wang et al., 2023b)	7B	71.0	48.3	44.6	43.6	96.8	82.0	**64.4**
Jina-V2 (Günther et al., 2023)	137M	50.3	54.5	37.9	38.9	93.5	74.0	58.2
Nomic-V1(Nussbaum et al., 2024)	137M	60.7	39.5	41.2	36.7	93.0	73.8	57.5
BGE-M3 (Chen et al., 2024)	568M	59.3	40.5	45.8	35.5	94.0	78.0	58.9
OpenAI-Ada-002	-	50.8	36.8	41.1	40.0	91.8	80.1	56.8
Our Extended Models								
$E5_{Base}$ + Tuning (4k)	110M	67.3	41.5	30.4	35.7	95.2	69.2	56.6
E5-$RoPE_{Base}$ + SelfExtend (4k)	110M	73.5	53.5	32.3	39.1	91.9	74.6	60.8
E5-Mistral + NTK (32k)	7B	**93.8**	**66.8**	**49.8**	**49.2**	**97.1**	**95.2**	**75.3**



(a) (b)

Figure 5: (a) Effects of different context window extension methods on $E5_{Base}$ and GTE_{Base}. We found that plug-and-play methods obtain similar scores, while further tuning yields the best results. Note that for *Tuning*, we report results based on PI by default, as evidenced by: (b) Further ablation of tuning setups. Tuning on PI consistently outperforms RP across both models.

including GP, RP, LPI and PCW strategies yield comparable results with no significant disparities. On the other hand, further tuning consistently yields additional performance gains for both models, across all target context lengths. Particularly noteworthy is GTE_{Base}, which showcases a substantial average score increase of approximately 5 points after further tuning. This suggests that freezing the original model weights and fine-tuning exclusively the added position embeddings can effectively extend the model's context window while strictly maintaining model's original ability.

RoPE-Based Models. Table 3 depicts the outcomes of E5-$RoPE_{Base}$ and E5-Mistral on each dataset of LONGEMBED after context window extension via PCW, GP, PI, SE and NTK. It is observed that

Model	Synthetic (Acc@1)		Real (nDCG@10)				Avg.
	Passkey	Needle	NQA	QMSum	SFD	2WmQA	
E5-RoPE$_{Base}$	*38.5*	*31.5*	*24.6*	*23.2*	*66.6*	*58.8*	*40.5*
+ PCW (4k)	42.5	50.8	25.1	34.9	**94.9**	69.3	52.9
+ GP (4k)	68.0	38.8	25.9	30.9	85.8	65.8	52.5
+ PI (4k)	68.3	36.0	25.9	30.8	84.9	65.3	51.9
+ SE (4k)	**73.5**	**53.5**	**32.3**	**39.1**	91.9	**74.6**	**60.8**
+ NTK (4k)	66.3	46.5	25.5	35.8	90.8	71.7	56.1
E5-Mistral	*71.0*	*48.3*	*44.6*	*43.6*	*96.8*	*82.0*	*64.4*
+ PCW (32k)	63.5	49.5	**59.3**	51.3	**97.3**	91.2	68.7
+ GP (32k)	81.0	48.8	37.0	42.9	90.6	88.1	64.7
+ PI (32k)	89.8	48.5	37.8	40.4	76.8	63.0	59.4
+ SE (32k)	90.8	52	49.3	48.7	97.2	**96.4**	72.4
+ NTK (32k)	**93.8**	**66.8**	49.8	**49.2**	97.1	95.2	**75.3**

Table 3: Results (%) of context window extension methods on E5-RoPE$_{Base}$ and E5-Mistral. *NQA*, *SFD*, *2WmQA* is short for *NarrativeQA*, *SummScreenFD*, *2WikiMultihopQA*, respectively.

RoPE-specific methods including NTK and SE yield significant improvements for both models across all datasets, surpassing PCW, PI and GP by a large margin.

5.4 Analysis

Tuning on PI vs. RP. Figure 5b compares further tuning on top of RP vs. PI. In the former approach, the initial 512 position embeddings are frozen while the remaining embeddings are tuned, whereas for the latter, the frozen / learnable embedding vectors are arranged in an interleaved manner. Our observations indicate that tuning applied to PI consistently produces superior results across both models. This superiority may be attributed to the fixed vectors acting as anchors, thereby preventing the learnable vectors from converging to suboptimal values.

RoPE vs. APE. We further discuss the potential of APE / RoPE-based models for context window extension. E5$_{Base}$ and E5-RoPE$_{Base}$ are selected as the comparison subjects thanks to their shared training process, training data, and comparable performance on BEIR and LongEmbed benchmarks. At each target context length ($\{1k, 2k, 4k\}$), we report the best scores achieved by each model on LongEmbed, as illustrated in Figure 6. Without requiring further training, E5-RoPE$_{Base}$ consistently demonstrates superior performance compared to E5$_{Base}$ across all target lengths. Furthermore, as the target window length increases, this superiority becomes more pronounced, even surpassing the fine-tuned version of E5$_{Base}$ by a large margin. This suggests that RoPE-based models can better extrapolate to to longer context. Consequently, we advocate for the use of RoPE in future embedding models.

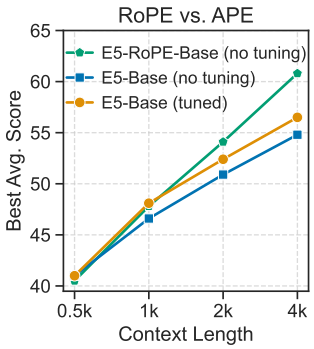


Figure 6: Best results achieved by extended versions of E5 / E5-RoPE.

6 Conclusion

This paper explores context window extension of existing embedding models. Through extensive experiments on our LongEmbed benchmark, we show that training-free context window extension strategies can effectively increase the input length of these models by several folds. Further, our analysis reveals the superiority of RoPE-based embedding models over APE-based ones in context window extension. Hence, we advocate for the use of RoPE for future embedding models.

LIMITATIONS

As a pioneering work in applying context window extension on embedding models, this paper is still limited in several aspects, particularly in that most of the context extension strategies explored in this paper are training-free. As evidenced by previous findings (Xiong et al., 2023; Fu et al., 2024; Zhang et al., 2024b; Yen et al., 2024), and the additional performance gain achieved via tuning on $E5_{Base}$ and GTE_{Base}, we believe further fine-tuning on top of plug-and-play methods can bring even better extension results. In the future, we will make comprehensive exploration of training-based context window extension for embedding models, especially for RoPE-based ones.

REFERENCES

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

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

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

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 2024.

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. Summscreen: A dataset for abstractive screenplay summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8602–8615, 2022.

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

David Chiang and Peter Cholak. Overcoming a theoretical limitation of self-attention. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7654–7664, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.527. URL https://aclanthology.org/2022.acl-long.527.

João Coelho, Bruno Martins, João Magalhães, Jamie Callan, and Chenyan Xiong. Dwell in the beginning: How language models embed long documents for dense retrieval. *arXiv preprint arXiv:2404.04163*, 2024.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41 (6):391–407, 1990.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. Longrope: Extending llm context window beyond 2 million tokens. *arXiv preprint arXiv:2402.13753*, 2024.

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

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6894–6910, 2021.

Tao Ge, Jing Hu, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. *arXiv preprint arXiv:2307.06945*, 2023.

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, et al. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents. *arXiv preprint arXiv:2310.19923*, 2023.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. URL https://www.aclweb.org/anthology/2020.coling-main.580.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Towards unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2(3), 2021.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376, 2023.

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

Greg Kamradt. Needle in a haystack - pressure testing llms. https://github.com/gkamradt/LLMTest_NeedleInAHaystack, 2023.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, 2020.

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

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, Daniel Haziza, Luca Wehrstedt, Jeremy Reizenstein, and Grigory Sizov. xformers: A modular and hackable transformer modelling library. https://github.com/facebookresearch/xformers, 2022.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

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

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. Mteb: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2014–2037, 2023.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative representational instruction tuning. *arXiv preprint arXiv:2402.09906*, 2024.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*, 2022.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human-generated machine reading comprehension dataset. 2016.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, et al. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9844–9855, 2022.

Zach Nussbaum, John X Morris, Brandon Duderstadt, and Andriy Mulyar. Nomic embed: Training a reproducible long context text embedder. *arXiv preprint arXiv:2402.01613*, 2024.

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

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

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Inbal Magar, Omri Abend, Ehud Karpas, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. Parallel context windows for large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6383–6402, 2023.

Anian Ruoss, Grégoire Delétang, Tim Genewein, Jordi Grau-Moya, Róbert Csordás, Mehdi Bennani, Shane Legg, and Joel Veness. Randomized positional encodings boost length generalization of transformers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1889–1903, 2023.

Jon Saad-Falcon, Daniel Y Fu, Simran Arora, Neel Guha, and Christopher Ré. Benchmarking and building long-context retrieval models with loco and m2-bert. *arXiv preprint arXiv:2402.07440*, 2024.

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. SCROLLS: Standardized CompaRison over long language sequences. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 12007–12021, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.823. URL https://aclanthology.org/2022.emnlp-main.823.

Jianlin Su. Understanding attention scaling from the perspective of entropy invariance. https://spaces.ac.cn/archives/8823, Dec 2021.

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

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=wCu6T5xFjeJ.

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

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Simlm: Pre-training with representation bottleneck for dense passage retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2244–2258, 2023a.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*, 2023b.

Suyuchen Wang, Ivan Kobyzev, Peng Lu, Mehdi Rezagholizadeh, and Bang Liu. Resonance rope: Improving context length generalization of large language models. *arXiv preprint arXiv:2403.00071*, 2024a.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. *Advances in Neural Information Processing Systems*, 36, 2024b.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, Song Han, and Maosong Sun. Infllm: Unveiling the intrinsic capacity of llms for understanding extremely long sequences with training-free memory. *arXiv preprint arXiv:2402.04617*, 2024.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. C-pack: Packaged resources to advance general chinese embedding. *arXiv preprint arXiv:2309.07597*, 2023.

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

Howard Yen, Tianyu Gao, and Danqi Chen. Long-context language modeling with parallel context encoding, 2024.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. Soaring from 4k to 400k: Extending llm's context with activation beacon. *arXiv preprint arXiv:2401.03462*, 2024a.

Yikai Zhang, Junlong Li, and Pengfei Liu. Extending llms' context window with 100 samples. *arXiv preprint arXiv:2401.07004*, 2024b.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. QMSum: A New Benchmark for Query-based Multi-domain Meeting Summarization. In *North American Association for Computational Linguistics (NAACL)*, 2021.

Dawei Zhu, Nan Yang, Liang Wang, Yifan Song, Wenhao Wu, Furu Wei, and Sujian Li. Pose: Efficient context window extension of llms via positional skip-wise training. In *The Twelfth International Conference on Learning Representations*, 2023.

A Training Details for E5-RoPE$_{\text{Base}}$

Params	Pre-training		Fine-tuning	
	E5$_{\text{Base}}$	E5-RoPE$_{\text{Base}}$	E5$_{\text{Base}}$	E5-RoPE$_{\text{Base}}$
learning rate	2×10^{-4}	2×10^{-4}	2×10^{-5}	2×10^{-5}
GPUs (V100)	32	32	8	8
warmup steps	1000	1000	400	400
max length	128	512	192	192
batch size	32k	16k	256	256
max steps	20k	20k	n.a.	n.a.
epochs	n.a.	n.a.	3	3
τ	0.01	0.01	0.01	0.01
α	n.a.	n.a.	0.2	0.2
weight decay	0.01	0.01	0.01	0.01
hard negatives	0	0	7	7
pos embedding	APE	RoPE	APE	RoPE

Table 4: Hyperparameters for contrastive pre-training and fine-tuning of E5$_{\text{Base}}$ and E5-RoPE$_{\text{Base}}$.

In this section, we describe the training details of E5-RoPE$_{\text{Base}}$. Our training procedure and data exactly follows that of E5 (Wang et al., 2022), where we first perform contrastive pre-training on their collected CCPairs, then perform fine-tuning on the concatenation of 3 datasets: MS-MARCO passage ranking (Nguyen et al., 2016), NQ (Karpukhin et al., 2020; Kwiatkowski et al., 2019), and NLI (Gao et al., 2021). Each example is paired with 7 hard negatives. We leverage the mined hard negatives and re-ranker scores from SimLM (Wang et al., 2023a) for the first two datasets. As the NLI dataset only provides 1 hard negative per example, we randomly sample 6 sentences from the entire corpus. xFormers (Lefaudeux et al., 2022) is used for memory efficient training. As presented in Table 4, training hyperparameters for E5$_{\text{Base}}$ and E5-RoPE$_{\text{Base}}$ are identical, except in two aspects:

- **Initialization.** Before contrastive pre-training, E5$_{\text{Base}}$ is initialized on BERT$_{\text{Base}}$ (Devlin et al., 2019), which employs absolute position embeddings (APE). For the initialization of E5-RoPE$_{\text{Base}}$, we simply replace the APE part of BERT$_{\text{Base}}$ with RoPE. It's worth noting that the BERT$_{\text{Base}}$ model after this replacement cannot function properly. We count on the subsequent pre-training phase to adapt the model to RoPE.
- **Pre-training length and batch size.** E5$_{\text{Base}}$ does not update its position embedding matrix during the training phase, i.e., it utilizes the same position embedding matrix as BERT$_{\text{Base}}$. This allows it to generalize to input sequences of up to 512 tokens, while being trained with a max training length of 192. As for E5-RoPE, replacing APE with RoPE during initialization prevents us from directly inheriting the original model's capability in handling 512 tokens. Consequently, in the pre-training phase of E5-RoPE, we set the maximum training length to 512, and reduce the batch size to 16k according to memory constraints.

Table 5 demonstrates results of E5$_{\text{Base}}$ and E5-RoPE$_{\text{Base}}$ on 15 publicly available BEIR tasks. We observe comparable overall scores between both models. This comparable performance, along with their shared training process and training data, facilitates fair comparison of APE and RoPE-based models's capabilities in length extrapolation. Note that the slight performance loss of E5-RoPE$_{\text{Base}}$ could possibly be attributed to the replacement of position embedding in the initialization phase, or the reduced batch size in the pre-training phase, as mentioned before.

B Implementation Details for Context Extension Strategies

This section describes implementation details for the explored context extension stratgies. For plug-and-play methods including PCW, RP, GP, PI, NTK and SE, Table 6 summarizes their hyperparameters under each condition.

Further Tuning. On top of PI and RP, we perform further tuning on both E5$_{\text{Base}}$ and GTE$_{\text{Base}}$, utilizing the fine-tuning dataset mentioned in Appendix A. Following the practice of PoSE (Zhu

Tasks	# W/Q.	# W/D.	$E5_{Base}$	$E5\text{-}RoPE_{Base}$
MS MARCO	6.0	56.0	41.8	42.4
Trec-Covid	10.6	160.8	69.6	73.3
NFCorpus	3.3	232.3	35.4	34.9
NQ	9.2	78.9	58.2	60.1
HotpotQA	17.6	46.3	69.1	61.0
FiQA	10.8	132.3	39.8	36.4
ArguAna	193.0	166.8	44.6	54.2
Touche-2020	6.6	292.4	26.4	26.6
CQADupStack	8.6	129.1	37.4	36.5
Quora	9.5	11.4	86.6	87.7
DBPedia	5.4	49.7	42.2	40.0
Scidocs	9.4	176.2	18.7	18.1
Fever	8.1	84.8	85.0	68.0
Climate-Fever	20.1	84.8	26.6	19.0
Scifact	12.4	213.6	72.0	71.0
Average	< 200	< 300	50.23	48.61

Table 5: Statistics and performance comparison of $E5_{Base}$ and $E5\text{-}RoPE_{Base}$ on 15 publicly available BEIR tasks. # W/Q. and # W/D. stands for word number per query and per document, respectively.

Extension	PCW & GP & RP & PI	NTK	SE
	GTE_{Base} & $E5_{Base}$		
512 -> 1,024	$L_o = 512, L_t = 1,024, s = 2$	-	-
512 -> 2,048	$L_o = 512, L_t = 2,048, s = 4$	-	-
512 -> 4,096	$L_o = 512, L_t = 4,096, s = 8$	-	-
	$E5\text{-}RoPE_{Base}$		
512 -> 1,024	$L_o = 512, L_t = 1,024, s = 2$	$\lambda = 3$ (10,000 -> 30,000)	$g = 3, w = 256$
512 -> 2,048	$L_o = 512, L_t = 2,048, s = 4$	$\lambda = 5$ (10,000 -> 50,000)	$g = 5, w = 128$
512 -> 4,096	$L_o = 512, L_t = 4,096, s = 8$	$\lambda = 10$ (10,000 -> 100,000)	$g = 9, w = 64$
	E5-Mistral		
4,096 -> 8,192	$L_o = 4,096, L_t = 8,192, s = 2$	$\lambda = 3$ (10,000 -> 30,000)	$g = 3, w = 2,048$
4,096 -> 16,384	$L_o = 4,096, L_t = 16,384, s = 4$	$\lambda = 5$ (10,000 -> 50,000)	$g = 5, w = 1,024$
4,096 -> 32,768	$L_o = 4,096, L_t = 32,768, s = 8$	$\lambda = 10$ (10,000 -> 100,000)	$g = 9, w = 512$

Table 6: Hyperparameters for plug-and-play context extension strategies.

et al., 2023), we manipulate position ids to simulate long training samples. Concretely, given an input document $\mathcal{D} = \{x_0, x_1, ..., x_{L_o-1}\}$ of original context length L_o, we introduce a skipping bias term u at the beginning of $\mathcal{D}$, transferring the original position ids $\mathcal{D}$ into $\{0, 1, ..., L_o - 1\}$ into $\{u, u + 1, ..., u + L_o - 1\}$. [4] For every piece of training data, u is re-sampled from the discrete uniform distribution $\mathcal{U}(\{0, 1, ..., L_t - L_o\})$. In this way, we ensure comprehensive coverage of target context window. The training procedure spans 3 epochs on 2 A100 GPUs, with a learning rate of $5e^{-4}$, a batch size of 512, and 100 steps for warmup. Other hyperparameters are same as Table 4.

Inference. In inference time, attention scaling (Su, 2021; Chiang & Cholak, 2022) is used by default for all tested models for better length extrapolation ability. Especially for GTE_{Base} and $E5_{Base}$ tuned on PI, we use the original position ids when input length not exceeds 512. This is achived by mapping the position ids $\{0, 1, ..., l\}$ into $\{0, s, ..., l \times s\}$, where s is the scaling factor, $l < 512$.

[4]The original practice of PoSE focuses on relative position, hence introduces bias terms at the middle of document $\mathcal{D}$. For APE-based models, we simply skips from the beginning.

Dataset Name	Source / Split	Query Example	Document Example
Narrative QA	- / test	Why is Bobolink eventually eager to help Martin?	The Project Gutenberg EBook of The Purple Cloud, by M.P. Shiel\n […] Title: The Purple Cloud\n\nAuthor: M.P. Shiel\n\nRelease Date: February 22, 2004, […]
QMSum	Scrolls / train + valid	The team wanted to understand how they could combine different linguistic features to make a more robust recognition model. They were […]	Project Manager: Can I close this ?\nUser Interface: Uh we don't have any changes , do we ?\nProject Manager: Oh , okay .\nUser Interface: So no . {vocalsound}\nProject Manager: {vocalsound} There we go . Okay , here we are again . Detailed design {disfmarker} oh , come on . Well {disfmarker} Ah {gap} s Forgot to insert the minutes […]
2WikiMultihopQA	LongBench / test	Where was the director of film The Central Park Five born	Passage 1:\nMargaret, Countess of Brienne\nMarguerite d'Enghien (born 1365 - d. after 1394), was the ruling suo jure Countess of Brienne and of Conversano, suo jure Lady of Enghien, and Lady of Beauvois from 1394 until an unknown date. […] Passage 2:\nNocher II, Count of Soissons\nNocher II (died 1019), Count of Bar-sur-Aube, Count of Soissons. He was the son of Nocher I, Count of Bar-sur-Aube. Nocher's brother Beraud (d. 1052) was Bishop of Soissons.Nocher became Count of Soissons, jure uxoris, upon his marriage to Adelise, Countess of Soissons. […]
SummScreenFD	Scrolls / valid	Penny gets a new chair, which Sheldon enjoys until he finds out that she picked it up from the street. He constantly pesters Penny to dispose of it, to no avail. Note: Melissa Rauch is absent in this episode.	[PREVIOUSLY_ON]\nYou make jumps you can't explain, Will. The evidence explains. Then help me find some evidence. I wouldn't put him out there! Should he get too close, I need you to make sure he's not out there alone. I don't think the Shrike killed that girl in the field. This girl's killer thought that she was a pig. You think this was a copycat? I think I can help good Will, see his face. Hello? They know.\n(gunshots)\nYou said he wouldn't get too close. See?\n(gunshots)\n(knocking)\nJack: We're here!\n(police radio chatter)\nWill: Could be a permanent installation in your Evil Minds Museum. […]
Passkey	- / -	what is the passkey for Kyree Mays?	[…] The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue.\nMalayah Graves's pass key is 41906. Remember it. 41906 is the pass key for Malayah Graves.\nThe sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. […]
Needle	- / -	What is the best thing to do in San Francisco?	Aaron Swartz created a scraped feed of the essays page. November 2021(This essay is derived from a talk at the Cambridge Union.) […] The best thing to do in San Francisco is eat a sandwich and sit in Dolores Park on a sunny day.\nThere's a narrow sense in which it refers to aesthetic judgements and a broader one in which it refers to preferences of any kind. […]

Figure 7: Source and examples for each dataset in LONGEMBED.

C FURTHER DETAILS ON LONGEMBED

Figure 7 presents source and examples for each dataset included in LONGEMBED. For QA datasets including NarrativeQA and 2WikiMultihopQA, we adopt their test splits. Note that for 2WikiMultihopQA, we adopt the length-uniformly sampled version from Bai et al. (2023b) to better assess the model's capabilities across various context lengths. For summarization datasets including QMSum and SummScreenFD, we adopt the version processed by SCROLLS (Shaham et al., 2022). Since SCROLLS does not include ground truth summarization in its test sets, we switch to validation set for these two datasets. Particularly for QMSum, as its validation set only have 60 documents, which is too small for document retrieval, we included the train set as well.