PEPE: Long-context Extension for Large Language Models via Periodic Extrapolation Positional Encodings

Jikun Hu[1], Dongsheng Guo[1*], Yuli Liu[1,2], Qingyao Ai[1,3], Lixuan Wang[1],
Xuebing Sun[5], Qilei Zhang[1], Quan Zhou[1,4], Cheng Luo[1,4]

[1]Quan Cheng Laboratory, Jinan, Shandong, China
[2]Qinghai University, Xining, Qinghai, China
[3]Tsinghua University, Beijing, China
[4]MegaTech.AI Inc., Beijing, China
[5]Dareway Software Co., Ltd., Jinan, Shandong, China

Abstract

Long-context extension seeks to expand the contextual window in pre-trained large language models (LLMs), allowing them to handle several multiples of their original training context lengths. The primary method for extending the window length involves expanding the initial positional encodings, such as interpolating and extrapolation new positions based on Rotary Position Embedding (RoPE). This expansion inevitably disrupts the positional encodings learned during pre-training, thereby affecting the attention allotment and introducing unseen positional encoding distributions. To address this issue, we propose a new extension strategy based on RoPE, namely **P**eriodic **E**xtrapolation **P**ositional **E**ncodings (PEPE). This strategy expands pre-trained high-dimensional components of positional encodings by replicating them in a periodic manner, thereby neither altering the learned positional encoding spaces nor introducing new positional encoding distributions. Experiments demonstrate that PEPE-based approaches can significantly improve long-context extension capabilities using just one-fourth the fine-tuning steps required by state-of-the-art methods. In addition, we analyze the characteristics of PEPE-based methods and the key parameters that contribute to their effectiveness. The code is publicly available [1].

1 Introduction

Nowadays, Transformer-based (Vaswani, 2017) large language models (LLMs) have experienced rapid advancement (Touvron et al., 2023a), showcasing impressive reasoning abilities that have significantly propelled progress in the field of natural language processing. Long-context extension (Chen et al., 2023) is a fundamental research area in the field of LLM, referring to the ability to process texts with substantially extended context lengths.

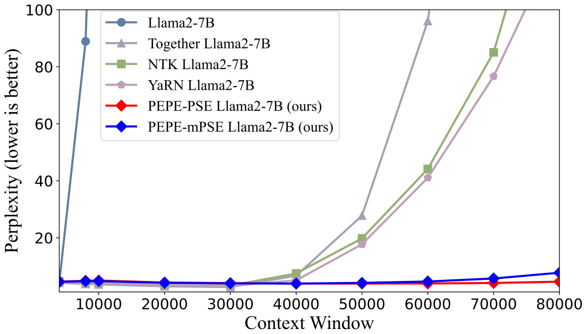


Figure 1: Perplexity comparison on Proof-pile (Azerbayev et al., 2022) documents between PEPE and other extension methods trained at a sequence length of 32k.

Extending the context window of LLMs basically requires augmenting the positional encodings, a critical step in improving the model's capability to handle longer sequences effectively. Rotary position embeddings (RoPE) (Su et al., 2021) have emerged as the leading positional encoding method for LLMs, as seen in widely used models such as Llama (Touvron et al., 2023a), PaLM (Chowdhery et al., 2023) and Qwen (Yang et al., 2024). Nevertheless, when the context length surpasses the length of the training data, a straightforward extrapolation based on RoPE fails to achieve the expected performance gains (Press et al., 2021). The key challenge is that the introduction of new position indices leads to an out-of-distribution (OOD) issue (Han et al., 2024). To alleviate the issues brought about by vanilla extrapolation, recent works have delved into the varying characteristics of positional encodings across different dimensional components. There has been a transition from linear interpolation (Chen et al., 2023) methods to non-uniform interpolation and extrapolation methods (Peng and Quesnelle, 2023; Peng et al., 2023). For instance, NTK-series methods (Peng and Quesnelle, 2023; Emozilla, 2023; Bloc97, 2023) address different dimensions of the positional encoding distinctly, while NTK-by-parts (Bloc97, 2023) and

[*] Corresponding author: `sr-guodsh@qcl.edu.cn`
[1] `https://github.com/JaxonHu-hub/PEPE`

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 21075–21085
November 4-9, 2025 ©2025 Association for Computational Linguistics

YaRN (Peng et al., 2023) divide positional encodings into three groups based on frequency, each adopting different approaches. Differentiating and processing positional encoding dimensions separately is an effective approach. Notably, YaRN has been adopted by several large language models (Liu et al., 2024; Yang et al., 2024), demonstrating its practical effectiveness in the community.

A key distinction across dimensions lies in the periodic properties exhibited within the pre-training window. From this perspective, the difference between the high-dimensional (low rotational frequency) and low-dimensional (high rotational frequency) components in RoPE resides in their ability to learn complete and sufficient sine-cosine periodic patterns (Liu et al., 2023). Extrapolating the low-dimensional components does not introduce new distributions in case of multiple complete sine-cosine cycles have already been learned, whereas the high-dimensional components behave in the opposite manner. Therefore, current methods predominantly employ interpolation for the high-dimensional components of positional encodings to mitigate severe OOD issues caused by extrapolation (Peng and Quesnelle, 2023; Peng et al., 2023; Emozilla, 2023; Bloc97, 2023). However, interpolation alters the rotational frequency, and learning new frequencies remains challenging, especially in the case of long extension length.

In this paper, according to the distinct periodic characteristics of low-dimensional versus high-dimensional components in positional encodings, we introduce PEPE (**P**eriodic **E**xtrapolation **P**ositional **E**ncodings), a new method for extending positional encodings. For high-dimensional components, PEPE leverages a cyclic mechanism that reutilizes the incompletely learned periods, facilitating an infinite extension without altering the pre-trained rotational frequency distribution. In contrast, for low-dimensional components, we retain the conventional direct extrapolation approach. This strategy not only ensures efficient learning but also enhances the effectiveness of long-range extrapolation, addressing the OOD challenge associated with extension context lengths. According to different cyclic modes, PEPE can be implemented in two ways: periodic shift extrapolation (PSE) and its mirrored approach (mPSE). Fig. 1 shows the comparison between the two modes of PEPE and other extrapolation methods. PEPE exhibits excellent stability in long-context expansion.

We summarize the contribution as follows:

(1) We propose a new long-context extension method, PEPE, based on the perspective of periodic integrity of high- and low-dimensional components of positional encodings, which significantly mitigates the OOD issue.
(2) We propose two cyclic mechanisms for the high-dimensional components, both of which demonstrate the effectiveness of PEPE.
(3) Experimental results show that PEPE achieves state-of-the-art performance in long-context extension with minimal training steps. It also demonstrates excellent stability and can be combined with other extrapolation techniques to further enhance its applicability.

2 Background and Related Work

2.1 Preliminary

Our work builds upon the RoPE method (Su et al., 2021), an enhanced approach to positional encoding that has since been widely adopted in LLMs. Formally, the encoding can be succinctly expressed for each token as follows (Ding et al., 2024):

$$\begin{aligned} R(m,\theta) = [&\cos(m\theta_0), \sin(m\theta_0), \cos(m\theta_1), \\ &\sin(m\theta_1), \ldots, \cos(m\theta_{d/2-1}), \sin(m\theta_{d/2-1})] \end{aligned} \tag{1}$$

where m denotes the index of token position, d represents the total dimensionality of positional encodings, and θ signifies the rotation frequency, defined as follows:

$$\theta_i = \frac{1}{base^{\frac{2i}{d}}}, \tag{2}$$

where $i \in [0, \frac{d}{2} - 1]$ denotes the paired dimension indices, and $base$ represents a frequency hyperparameter. To more clearly represent the position encodings across different dimensions, we define $\delta \in [0, d)$ as the dimension index, and the encodings within δ dimension as $f_\delta(m, \theta_i)$.

Distinguishing high-dimensional and low-dimensional components is based on whether there are n complete sine-cosine periods for different dimensions. We define the distinction point $\hat{\delta}$ such $m\theta_i = 2n\pi$. According to Eq. 2, this yields $m\frac{1}{base^{\frac{\hat{\delta}}{d}}} = 2n\pi$. Solving for $\hat{\delta}$, we obtain:

$$\hat{\delta} = \lceil d \log_{base} \frac{m}{2n\pi} \rceil. \tag{3}$$

Considering Llama2 models (Touvron et al., 2023b), let $base$=10000, and d=128 as defaults and take m=4096 as an example. Assuming n=1, corresponding to one complete sine-cosine period, we can calculate that the distinction point $\hat{\delta}$=92 according to Eq. 3. Thus, indices $\delta \leq 92$ correspond to the low-dimensional components, whereas indices $\delta > 92$ are associated with the high-dimensional components. Additionally, in the long-context extension task, we denote the original pre-trained context length as L, the extension length as L'.

2.2 Linear positional interpolation

Linear positional interpolation (PI) (Chen et al., 2023) is a straightforward manner to implement linear interpolation to compresses the positional indices of RoPE proportionally based on the extension ratio $s = \frac{L'}{L}$. PI reduces the distance between adjacent tokens across all dimensions at a certain ratio. The newly introduced compact positional encoding introduces challenges for the model in distinguishing subtle differences between adjacent positions, particularly under high scaling ratios where performance deteriorates significantly.

2.3 Non-uniform extension methods

NTK-series methods (Peng and Quesnelle, 2023; Emozilla, 2023; Bloc97, 2023) are a set of non-linear interpolation and extrapolation methods. Instead of directly adopting a fixed radio, they distribute interpolation pressure across different dimensions to mitigate the crowded-positions issue in PI based on Neural Tangent Kernel (NTK) theory (Jacot et al., 2018). According to the density characteristics of positional information in different dimensions, they insert less in lower dimensions and more in higher dimensions, resulting in both interpolation and extrapolation. NTK-by-parts (Bloc97, 2023) first divides dimensions into three dimension groups, each with a different interpolation strategy. It uses PI for high-dimensions while low-dimensions undergo extrapolation, and use NTK-aware (Peng and Quesnelle, 2023) in-between. YaRN (Peng et al., 2023) achieves better extrapolation performance than NTK-by-parts by incorporating attention scaling technology. From a periodicity standpoint, since low-dimensional extrapolation does not generate new periodic structures while high-dimensional spaces tend to do the opposite, these methods may naturally conform to this behavior. LongRoPE (Ding et al., 2024) determines the optimal dimensional split points via a search-based approach and applies distinct interpolation methods across three dimensional groups, similar to NTK-by-parts and YaRN.

Other long-context window extension methods mainly include memory-retrieval approaches (Borgeaud et al., 2022; Tworkowski et al., 2023; Wang et al., 2023) and attention manipulating mechanisms (Ratner et al., 2022; Han et al., 2024; Xiao et al., 2023). These methods serve as complements to the original LLM architecture and can be applied in conjunction with the RoPE-based position extension route (Ding et al., 2024). We focus on achieving long-context extension efficiently with minimal fine-tuning at short-context length.

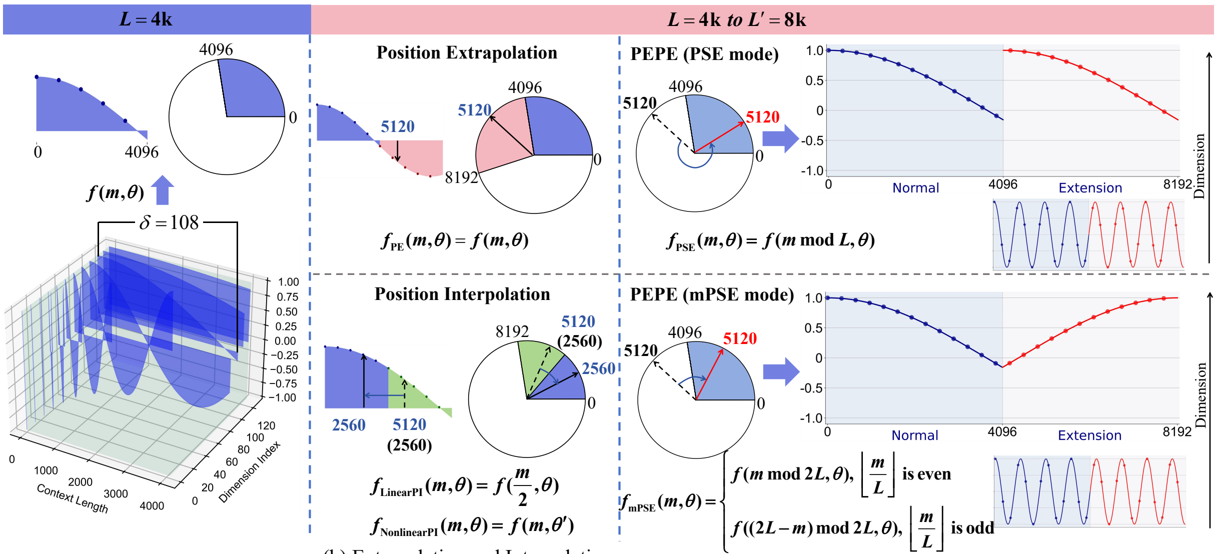
3 Periodic Extrapolation Positional Encodings

Motivated by the periodic incompleteness of high-dimensional components in positional encodings inevitably leads to OOD issues, we present Periodic Extrapolation Positional Encodings (PEPE), a positional encoding method that extends long-context sequences by periodically replicating pre-trained high-dimensional components.

3.1 Periodic View on OOD Problems

Current position encoding interpolation and extension methods inevitably face OOD issues, and we examine this problem from the perspective of periods as follows. In RoPE, as the dimensionality increases, the number of sine-cosine cycles gradually decreases until it becomes impossible to complete a full cycle. Specifically, according to Eq. 2, as the dimensionality increases, θ gradually decreases, leading to a reduction in the rotation speed. Consequently, at a certain dimension, it becomes impossible to complete a full sine-cosine cycle. Fig. 2(a) illustrates this phenomenon with δ=108 as an example, where the current dimensional position encoding is denoted as $f = (m, \theta)$.

Taking the extension length from L=4k to L'=8k as an example, Fig. 2(b) illustrates common position extrapolation (PE) and interpolation (PI) methods. As shown that, in the case of incomplete sine-cosine cycles (δ=108), PE is extrapolated along the original sine-cosine cycles, as $f_{\text{PE}}(m, \theta) = f(m, \theta)$, leading to untrained position encoding patterns (see the top of Fig. 2(b)). PI decreases the rotation frequency on existing periods, which essentially compresses the periodic space. As Fig. 2(b) shows, linear PI compresses the original length to half of its original space, as $f_{\text{LinearPI}}(m, \theta) = f(\frac{m}{2}, \theta)$. For example, the extended token position m=5120 corresponds to the original token position m=2560. Non-uniform PI



(a) RoPE in Pre-trained Length

(b) Extrapolation and Interpolation Based on RoPE

(c) Periodic Extrapolation Positional Encodings Based on RoPE

Figure 2: An overview of different long-context extension methods based on RoPE, taking $\delta = 108$ as an example. (a) illustrates the encoding of RoPE within the pre-training window length. (b) shows the encoding resulting OOD phenomena of the widely used positional extrapolation and interpolation methods. (c) presents two modes of our Periodic Extrapolation Positional Encodings (PEPE) method.

achieves dynamic frequency interpolation by modifying θ, as $f_{\mathrm{NonlinearPI}}(m,\theta) = f(m,\theta')$. A common approach is to increase the $base$, and further obtain θ' according to Eq. 2 (Roziere et al., 2023). Actually, both of linear and non-linear PI methods reduce the distance between adjacent tokens, resulting in disrupting the original distance patterns among tokens.

This suggests that, in the case of high-dimensional positional components, PE tends to extrapolate beyond the pre-training range, whereas PI introduces denser positional encodings. Both methods exhibit OOD limitations. Accordingly, an ideal approach should maintain the integrity of the pre-trained positional encoding distribution in high dimensions. Inspired by this, we introduce PEPE, a straightforward yet effective method that extends positional encodings by treating incomplete cycles as independent units and repeating them, instead of enforcing full sinusoidal periodicity. Fig. 2(c) illustrates the different encoding extrapolation methods used by the PEPE approach for low and high dimensions. PEPE rotates the positional encodings that have been extrapolated beyond the pre-trained distribution back into the original distribution through shift operations. PEPE has two typical characteristics: First, in the extension window, PEPE is a purely extrapolation method that does not perform interpolation in any dimension, thereby not altering the periodic sampling frequency. Second, by performing a replication extrapolation operation on the high-dimensional components of positional encodings, it avoids the additional distribution introduced by general extrapolation.

There are two modes for PEPE, as shown in Fig. 2(c). The first mode involves periodic shift extrapolation (PSE) of high-dimensional components. The second mode involves flipping the adjacent high-dimensional components horizontally before shifting them, *i.e.,* mirrored periodic shift extrapolation (mPSE). The latter method can maintain the continuity of positional encodings but needs more fine-tuning steps to learn the mirrored ways.

3.2 Periodic shift extrapolation

Based on the preceding analysis, we only need to apply periodic shift handling to dimensions higher than distinction dimensional point $\hat{\delta}$, while direct extrapolation is used for the lower-dimensional component. Formally, our periodic shift extrapolation encodings with δ can be represented as:

$$f_{\mathrm{PSE}}(m,\theta,\delta) = \begin{cases} f(m,\theta), & \text{if } \delta \le \hat{\delta} \\ f(m \mod L,\theta), & \text{if } \delta > \hat{\delta} \end{cases} \tag{4}$$

According to Eq. 4, $\hat{\delta}$ is a critical parameter that determines from which δ to start applying the PSE. From the perspective of sine-cosine periodicity analysis, we adopt the first full sine-cosine period dimension (n=1) for $\hat{\delta}$ as default.

Another critical parameter is $\hat{m}$, which indicates from which token to start applying the PSE. In Fig. 2(c), we begin using PSE from the extrapolation portion ($\hat{m}$=L) as default. Actually, considering the issue of attention reallocation for the extrapolated context, advancing $\hat{m}$ would be a better choice. Therefore, the complete periodic shift extrapolation can be represented as:

$$f_{\text{PSE}}(m, \theta) = \begin{cases} f(m, \theta), & \text{if } m \leq \hat{m} \\ f_{\text{PSE}}(m, \theta, \delta), & \text{otherwise} \end{cases} \tag{5}$$

PSE has a notable characteristic that there are discontinuities between adjacent periods (see m=4096 at top of Fig. 2(c)). It may increase the difficulty of learning attention for tokens on either side of the discontinuity. To address this potential issue, we propose the mirrored PSE mode.

3.3 Mirrored periodic shift extrapolation

In response to the issue of discontinuous position encodings at the periodic junctions in PSE, we propose its mirrored mode, namely mPSE. The bottom of Fig. 2(c) shows the mPSE mode, which introduces a mirrored pattern that flips the periodic shifts of odd-numbered cycles left and right, based on PSE, ensuring continuous transitions between adjacent periods. Formally, for high-dimensional components, mPSE can be represented as:

$$f_{\text{mPSE}}^{\text{high}}(m, \theta) = \begin{cases} f(m \mod 2L, \theta), & \text{if } \lfloor \frac{m}{L} \rfloor \text{ is even,} \\ f((2L - m) \mod 2L, \theta), & \text{otherwise,} \end{cases} \tag{6}$$

where $\lfloor \frac{m}{L} \rfloor$ denotes the number of periodic translation. Similar to Eqs. 4 and 5 in PSE, mPSE also has two important hyperparameters $\hat{\delta}$ and $\hat{m}$.

The two modes, PSE and mPSE, both avoid OOD operations on the high-dimensional components of encodings through copy and shift operations, thereby demonstrating a rapid fitting capability during fine-tuning. Between the two, PSE is simpler to learn, whereas mPSE demonstrates superior performance in terms of attention continuity.

4 Experiments

To verify the effectiveness of our method, we first compared common extension methods on three tasks (Sec. 4.2). Then, we perform ablations of the hyperparameters that affect PEPE's performance (Sec. 4.3). Additionally, we analyze two implementation modes of PEPE (Sec. 4.4). More details of experiments can be found in the Appendix.

4.1 Setup

Baselines. The experiments are conducted on Llama2-7B under three training sequence length settings: 8k, 16k, and 32k. We compare several commonly used extension techniques with our methods, including PI (Chen et al., 2023), NTK (Peng and Quesnelle, 2023), and YaRN (Peng et al., 2023), using their official training configurations. We utilize the "togethercomputer/Llama-2-7B-32k" (Together.ai, 2023) model with PI applied, without further fine-tuning.

Training Details. Following YaRN (Peng et al., 2023), we use a learning rate of 2e-5 with linear decay and a global batch size of 64 on PG19 dataset (Rae et al., 2019). We set $\hat{\delta}$=92 (n=1) and $\hat{m}$=1.5k for both PSE and mPSE modes by default. Additionally, we incorporate the attention scaling technique from YaRN during the training of PEPE. Notably, PEPE requires only 100 training steps, while other methods like YaRN require four times as many steps to converge. All experiments are conducted on 8 A100 40GB GPUs. Due to GPU memory limitations, we are only able to test up to an extension context length of 80k.

4.2 Main Results

Similarly to LongRoPE (Ding et al., 2024), we conduct a comprehensive evaluation of PEPE's effectiveness by assessing its performance across three key aspects: (1) perplexity on long documents, (2) passkey retrieval task, and (3) LLM benchmarks.

Perplexity (PPL). We begin by comparing various state-of-the-art RoPE-based sequence extension methods in terms of perplexity. The evaluation is carried out on the Proof-pile dataset (Azerbayev et al., 2022), which contains a substantial amount of long-text content. Following YaRN (Peng et al., 2023), we select 10 samples from the Proof-pile dataset with sequence lengths exceeding 80k tokens for each run. These sequences are then progressively truncated from 4k to 80k tokens to assess model performance across varying context lengths. All perplexity are computed using a sliding window (Press et al., 2021) of size 256.

Table 1 summarizes the comparison results. We evaluate the performance across different training length configurations, specifically 8k, 16k, and 32k, and also report the corresponding number of training steps required. Experimental results show that PEPEs offer two key advantages: (1) PEPEs exhibit excellent stability in terms of PPL across different

Table 1: Perplexity results using PI (Chen et al., 2023), NTK (Peng and Quesnelle, 2023), YaRN (Peng et al., 2023) and PEPE extension methods on Proof-pile dataset (Azerbayev et al., 2022), evaluated on the Llama2-7B model. We set $\hat{\delta}$=92 and $\hat{m}$=1.5k for both PSE and mPSE.

Training Length	Model Name	Extension Method	Training Steps	Evaluation Context Length								
				4k	10k	20k	30k	40k	50k	60k	70k	80k
4k	Llama2-7B	-	-	4.16	$>10^2$	$>10^3$	$>10^3$	$>10^3$	$>10^3$	$>10^3$	$>10^3$	$>10^3$
32k	Together-Llama2-7B	PI	-	**4.21**	**3.66**	**2.98**	**2.73**	6.70	27.78	96.05	$>10^2$	$>10^2$
	NTK Llama2-7B	NTK	400	4.76	4.14	3.40	3.25	7.49	19.79	44.15	85.05	$>10^2$
	YaRN Llama2-7B	YaRN	400	4.70	4.11	3.38	3.11	5.08	17.73	41.07	76.67	$>10^2$
	PEPE-PSE Llama2-7B	PSE	**100**	5.07	4.49	3.74	3.44	3.22	3.10	3.01	2.95	2.91
	PEPE-mPSE Llama2-7B	mPSE	**100**	4.96	4.40	3.64	3.35	**3.15**	**3.02**	**2.92**	**2.87**	**2.83**
16k	NTK Llama2-7B	NTK	400	5.37	4.78	7.36	53.11	$>10^2$	$>10^2$	$>10^2$	$>10^2$	$>10^2$
	YaRN Llama2-7B	YaRN	400	**4.79**	**4.19**	5.38	45.44	$>10^2$	$>10^2$	$>10^2$	$>10^2$	$>10^2$
	PEPE-PSE Llama2-7B	PSE	**100**	5.62	5.09	**4.29**	**3.98**	**3.74**	**3.63**	**3.55**	**3.53**	**3.50**
	PEPE-mPSE Llama2-7B	mPSE	**100**	5.91	5.36	4.51	4.19	3.95	3.84	3.77	3.76	3.76
8k	NTK Llama2-7B	NTK	400	5.04	7.66	$>10^2$	$>10^2$	$>10^3$	$>10^3$	$>10^3$	$>10^3$	$>10^3$
	YaRN Llama2-7B	YaRN	400	**4.83**	7.08	$>10^2$	$>10^2$	$>10^3$	$>10^3$	$>10^3$	$>10^3$	$>10^3$
	PEPE-PSE Llama2-7B	PSE	**100**	10.85	9.52	8.00	7.52	7.18	7.26	7.30	7.45	7.64
	PEPE-mPSE Llama2-7B	mPSE	**100**	6.39	**5.87**	**4.95**	**4.66**	**4.49**	**4.50**	**4.55**	**4.75**	**5.07**

training lengths and under long-sequence extrapolation scenarios. Even when trained on sequences of length 8k, they can stably extrapolate to a context length of 80k during inference. (2) Compared to methods such as YaRN, PEPE requires only one-fourth of the training steps to achieve outstanding long-sequence extension performance.

Passkey Retrieval. The Passkey Retrieval task (Mohtashami and Jaggi, 2023) evaluates the ability to locate a specific key embedded within an extremely long context. We conduct 10 iterations of the passkey retrieval task using a training length of 32k, with input lengths ranging from 4k to 70k tokens, to assess performance across increasing context lengths. As shown in Table 2, other methods like PI, NTK, and YaRN exhibit a sharp decline in performance beyond the original training window. In contrast, PEPE maintains strong extrapolation capabilities even with only 100 fine-tuning steps. Furthermore, increasing the number of training steps significantly improves the stability and adaptability of both PSE and mPSE, as shown in last two rows of Table 2.

Standard LLM benchmarks. We conduct a systematic evaluation on the Hugging Face Open LLM Leaderboard (Fourrier et al., 2024) focusing on the original 4k context window with 32k training length. This benchmark includes four tasks: 25-shot ARC-Challenge (Clark et al., 2018), 10-shot HellaSwag (Zellers et al., 2019), 5-shot MMLU (Hendrycks et al., 2020), and 0-shot TruthfulQA (Lin et al., 2021), which are commonly used to assess language understanding and reasoning capabilities. As shown in Table 3, extending the context length from the original Llama2-7B leads to a slight performance drop across multiple tasks for all extrapolation methods. Overall, PEPE performs at a level that is competitive with other extension methods. Additionally, we conduct an extra evaluation of PEPE under m=4k, where the positional encodings within the 4k context length are left unchanged. The results show slight improvements over the 1.5k baseline across all four metrics.

Table 2: Passkey retrieval accuracy of long-context LLMs under various positional extension methods, with sequence lengths ranging from 4k to 70k tokens.

Extension Method	Training Steps	Evaluation Context Length							
		4k	10k	20k	30k	40k	50k	60k	70k
RoPE	-	1	0	0	0	0	0	0	0
PI	-	1	1	1	1	0	0	0	0
NTK	400	1	1	0.9	0.4	0	0	0	0
YaRN	400	1	1	1	1	0	0	0	0
PSE	100	1	0.8	1	0.9	0.8	1	0.8	1
mPSE	100	1	0.9	0.9	0.8	0.5	0.7	0.8	0.2
PSE	150	1	1	0.9	1	1	1	1	0.9
mPSE	250	1	1	1	1	0.8	1	0.7	1

4.3 Ablation Studies

The parameters that may influence PEPE's performance include its key internal components: the

Table 3: Comparison of different extension methods on standard LLM benchmarks, evaluated within a 4k context length. ‘ARC.’, ‘Hel.’, and ‘Tru.’ denote ARC-Challenge, HellaSwag and TruthfulQA, respectively.

Methods	ARC.	Hel.	MMLU	Tru.
Llama2-7B	49.91	58.99	32.05	45.71
PI	43.26	57.32	31.89	**44.89**
NTK	47.95	56.27	29.73	36.60
YaRN	48.46	56.84	**32.91**	41.25
PSE (m=1.5k)	47.01	56.19	31.35	39.08
mPSE (m=1.5k)	46.93	56.28	30.38	38.38
PSE (m=4k)	**48.89**	**57.54**	31.41	42.39
mPSE (m=4k)	48.55	56.38	31.09	39.99

dimension distinct point $\hat{\delta}$ and the starting token position $\hat{m}$, as well as the number of training steps, the adjustable $base$ value, and whether or not to incorporate the attention scaling technique. In the following, we present a detailed analysis of each parameter under the default configuration: n=1 ($\hat{\delta}$=92) , $\hat{m}$=4k, 100 training steps, $base$=10,000 and without using attention scaling. PSE and mPSE are both trained on sequences of length 32k.

Analysis on $\hat{\delta}$. The value of $\hat{\delta}$ affects the starting dimension of PEPE's cyclic shift. To evaluate its impact, we measure perplexity across $\hat{\delta}$ values ranging from n=0 to ∞ sine-cosine cycle for both PSE and mPSE, respectively. n=0 indicates that no dimensions are considered while n=∞ denotes that all dimensions are involved. The results, as shown in Table 4, indicate that selecting one full cycle as the $\hat{\delta}$ is not necessarily optimal. From the longer length perspective, selecting $\hat{\delta}$ value corresponding to around 0.5 cycle during training is a more favorable choice. This implies that a lower-dimensional starting point can bring about slight performance improvement by reducing the number of dimensions directly extrapolation. The context expansion performance drops after exceeding the fine-tuned context length, when high- and low-dimensional components are not distinguished (n=∞). When n=0, *i.e.*, without using PEPE and continuing to use vanilla RoPE, the performance deteriorates significantly beyond a reasoning length of 32k.

Analysis on $\hat{m}$. The default setting is m=4k, which means the positional encodings within the pre-learned context window remain unchanged. We conduct experiments by gradually decreasing m from 4k to 1k. The experimental results shown in Table 5 indicate that integrating PEPE earlier can improve the perplexity performance. This is because the model learns more periodic patterns within the pre-training window, which helps in extending to longer context lengths. However, if the integration occurs too early (*e.g.*, m=1k), it may lead to performance degradation. This could be due to the early integration significantly increasing the perturbation to the positional encodings within the pre-training window. Additionally, as indicated by Eq. 3, reducing m shifts the dimensional distribution toward lower dimensions, causing more dimensions to be involved in PEPE and thus exacerbating this negative effect.

Table 4: Perplexity of PSE and mPSE across varying numbers n of sine-cosine cycles starting from different dimensions. ∞ denotes that all dimensions are involved.

Extension Method	n	Evaluation Context Length					
		4k	10k	20k	40k	60k	80k
RoPE	0	4.76	8.11	22.4	60.9	109	160
PSE	∞	4.70	5.54	5.01	6.76	49.2	161
	2	**4.49**	4.85	4.14	3.74	3.83	4.91
	1	4.65	5.01	4.31	3.95	3.98	4.61
	0.5	4.50	**4.72**	**4.05**	**3.68**	**3.69**	**4.15**
mPSE	∞	4.57	5.64	5.32	7.11	25.9	77.6
	2	**4.50**	**4.73**	**4.08**	**3.70**	4.05	7.17
	1	4.56	4.80	4.20	3.94	4.67	7.75
	0.5	4.55	4.78	4.12	3.81	**4.01**	**4.54**

Table 5: Perplexity at different starting positions $\hat{m}$.

Extension Method	$\hat{m}$	Evaluation Context Length					
		4k	10k	20k	40k	60k	80k
PSE	4k	**4.65**	5.01	4.31	3.95	3.98	4.61
	3k	5.18	4.95	4.31	3.85	**3.79**	**4.09**
	2k	4.85	4.49	3.86	**3.50**	3.86	6.59
	1.5k	4.86	4.40	3.72	**3.50**	6.30	18.2
	1k	4.80	**4.30**	**3.65**	4.06	16.0	56.6
mPSE	4k	4.99	4.81	4.11	3.63	3.58	3.76
	3k	4.57	4.58	4.00	3.53	3.42	3.52
	2k	**4.80**	**4.39**	3.70	3.33	3.71	6.95
	1.5k	5.02	4.40	**3.66**	**3.22**	**3.07**	**3.06**
	1k	4.89	4.41	3.78	3.45	6.96	25.6

Analysis on training steps. We separately evaluate PSE and mPSE over training steps ranging from 0 to 400. The experimental results are shown in Fig. 3. After more than 50 training steps, both PSE and mPSE achieve stable PPL under the 80k extrapolation range. The trend suggests that the model has the potential to perform well beyond 80k. Notably, mPSE achieves better performance with more training steps.

Analysis on $base$ value. Increasing the $base$ value

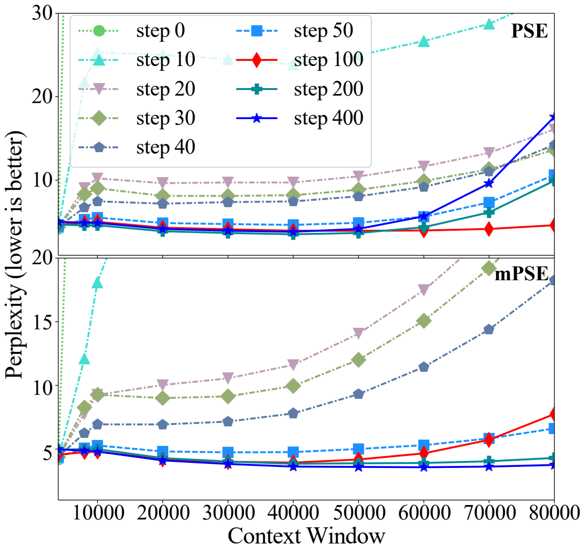


Figure 3: Perplexity comparison of PSE and mPSE over training steps from 0 to 400 on Proof-pile documents.

Table 6: Perplexity of PSE and mPSE with RoPE *base* adjustment, *base*=10,000 as default.

Extension Method	*base*	Evaluation Context Length					
		4k	10k	20k	40k	60k	80k
PSE	10^4	4.65	5.01	4.31	3.95	**3.98**	**4.61**
	10^5	**4.55**	**4.09**	3.43	3.33	12.8	46.8
	10^6	4.59	**4.09**	**3.40**	**3.11**	8.82	31.8
mPSE	10^4	**4.56**	4.80	4.20	3.94	4.67	7.75
	10^5	4.61	4.12	3.42	2.98	**3.19**	**5.13**
	10^6	4.58	**4.05**	**3.36**	**2.97**	7.84	28.7

of RoPE is a commonly used strategy in long-context extension techniques (Roziere et al., 2023; Liu et al., 2023). Therefore, we investigate the combined effect of *base* and PEPE. We first evaluate perplexity under different *base* values, and the results are shown in Table 6. The results indicate that increasing the *base* value can improve performance within a certain extrapolation range. However, when extrapolating to much longer sequence lengths, performance gradually degrades, as altering the RoPE rotation frequency disrupts the pre-trained positional encoding patterns.

Ablation of attention scaling. Further, we examine the effect of integrating the attention scaling technique from YaRN (Peng et al., 2023) into PEPE. The experiments include both perplexity evaluation and passkey retrieval task under m=1.5k. As shown in Table 7, integrating attention scaling into PEPE can further improve both perplexity and passkey retrieval performance, building upon PEPE's inherent long-context extension capability. This demonstrates that PEPE has good compatibility with other techniques and can work synergistically with them.

Table 7: Ablation study of attention scaling on perplexity evaluation and passkey retrieval.

Extension Method	Attn. Scaling	Evaluation Context Length					
		4k	10k	20k	40k	60k	80k
		Perplexity					
PSE	×	**4.86**	**4.40**	**3.72**	3.50	6.30	18.2
	✓	5.07	4.49	3.74	**3.22**	**3.01**	**2.91**
mPSE	×	5.02	**4.40**	3.66	3.22	3.07	3.06
	✓	**4.96**	**4.40**	**3.64**	**3.15**	**2.92**	**2.83**
		Passkey retrieval					
PSE	×	**1.00**	**1.00**	0.80	**0.95**	0.45	0.25
	✓	0.95	**1.00**	**0.95**	0.80	**0.90**	**0.90**
mPSE	×	**1.00**	0.85	0.90	0.85	**0.90**	0.60
	✓	**1.00**	**0.95**	**1.00**	**1.00**	0.80	**0.70**

4.4 Discussion Between PSE and mPSE

Based on the experiments in Secs. 4.2 and 4.3, we can draw the following comparative conclusions regarding PSE and mPSE: (1) mPSE requires more training steps to reach optimal performance compared to PSE. Because mPSE applies a mirroring operation before performing period extension which needs more training steps to fully learn the periodic patterns. (2) mPSE demonstrates superior overall stability compared to PSE, as evidenced by its lower perplexity on long sequences and smaller performance drops when sequence length exceeds the injection position point. PSE exhibits abrupt changes in attention scores at injection points ($\hat{m}$ and its multiples), whereas mPSE demonstrates smoother transitions. This difference is likely the primary factor contributing to mPSE's enhanced stability during extrapolation.

5 Conclusion

This paper presents PEPE, a novel positional encoding extension method designed to address the OOD challenges. From a periodic perspective, by applying shift operations on high-dimensional components of positional encodings, PEPE avoids introducing new positional information while maintaining compatibility with pre-trained models. Based on PEPE, we develop two practical variants, PSE and mPSE, both of which demonstrate strong stability under large extrapolation ratios. The results show that PEPE achieves superior perplexity performance and retrieval accuracy compared to current methods. Looking forward, PEPE opens new possibilities for efficient context length extension in LLMs, with potential applications in long-text understanding, agent reasoning, and beyond.

6 Limitations

Our study has two main limitations. First, due to GPU memory constraints, we only conduct comparisons on Llama2 within an 80k context length extension. Based on the observed performance trends, 80k does not appear to be the upper limit of PEPE's extension capability, and further investigation at longer lengths is needed. Second, within the fine-tuning context window range, compared to methods like PI, the PEPE method does not demonstrate significant performance gains. PEPE focuses on achieving a large-scale context window extension under limited fine-tuning lengths. If sufficient computational resources are available, fine-tuning at the target extended context length would be a better alternative than PEPE.

7 Acknowledgments

This work is supported by the Department of Science & Technology of Shandong Province (No. SYS202201) and the Quan Cheng Laboratory.

References

Z. Azerbayev, E. Ayers, and B. Piotrowski. 2022. Proof-pile. https://github.com/zhangir-azerbayev/proof-pile.

Bloc97. 2023. Add NTK-Aware interpolation "by parts" correction. Accessed: 2025.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

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

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. PaLM: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

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

Emozilla. 2023. Dynamically scaled RoPE further increases performance of long context llama with zero fine-tuning. Accessed: 2025.

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. 2024. Open LLM leaderboard v2. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard.

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. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3991–4008.

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*.

Arthur Jacot, Franck Gabriel, and Clément Hongler. 2018. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, 31.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Xiaoran Liu, Hang Yan, Shuo Zhang, Chenxin An, Xipeng Qiu, and Dahua Lin. 2023. Scaling laws of RoPE-based extrapolation. *arXiv preprint arXiv:2310.05209*.

Amirkeivan Mohtashami and Martin Jaggi. 2023. Random-access infinite context length for transformers. *Advances in Neural Information Processing Systems*, 36:54567–54585.

Bowen Peng and Jeffrey Quesnelle. 2023. NTK-aware scaled RoPE allows llama models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation. Accessed: 2025.

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

Ofir Press, Noah A Smith, and Mike Lewis. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*.

Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, and Timothy P Lillicrap. 2019. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*.

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Inbal Magar, Omri Abend, Ehud Karpas, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2022. Parallel context windows improve in-context learning of large language models. arxiv.

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

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

Together.ai. 2023. togethercomputer/llama-2-7b-32k.

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

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

Szymon Tworkowski, Konrad Staniszewski, Mikołaj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Miłoś. 2023. Focused transformer: Contrastive training for context scaling. *Advances in Neural Information Processing Systems*, 36:42661–42688.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. 2023. Augmenting language models with long-term memory. *Advances in Neural Information Processing Systems*, 36:74530–74543.

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

Mingyu Xu, Xin Men, Bingning Wang, Qingyu Zhang, Hongyu Lin, Xianpei Han, et al. 2024. Base of rope bounds context length. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

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

A Appendix

A.1 Additional analysis on $base$ value

Refer to (Xu et al., 2024), we analyze the attention scores with respect to relative distances under similar tokens for both PSE and mPSE. The results are shown in Fig. 4. It can be observed that: (1) Increasing the $base$ value significantly amplifies the attention magnitude. (2) Neither PSE nor mPSE introduces excessive long-distance decay, which is attributed to the positional shift strategy of high dimensions.

According to these two characteristics, we further evaluate the performance by increasing the $base$ in the passkey retrieval task. We select the two best-performing PEPE models: PSE, which was fine-tuned for 100 steps, and mPSE, which was fine-tuned for 400 steps. Other parameters are the same as those used in the default ablation study in the main text. As shown in Table 8, with default $baset$, PEPE methods exhibit relatively limited performance on this structured retrieval task. When combined with $base$ increasing to 10^5, their performance improves significantly. Performance on the passkey retrieval task can be enhanced by appropriately increasing the base, especially when using the mPSE method. Additionally, we find that $\hat{m}$ also has a notable impact on the performance of the passkey retrieval task, and we discuss this further below.

A.2 Impact of $\hat{m}$ on passkey retrieval efficiency

We conduct experiments with $\hat{m}$ in the range of 1.5k to 4k. The experimental results shown in Table 9 indicate that by reducing the value of $\hat{m}$, the performance on the passkey retrieval task can be significantly improved. Moreover, a larger number of fine-tuning steps yields a substantial gain in

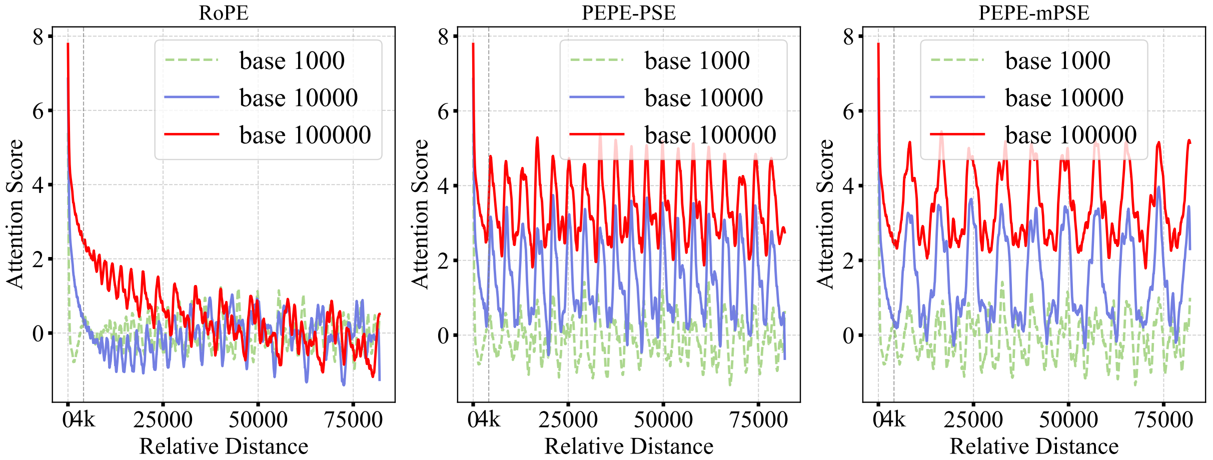


Figure 4: The attention scores of similar tokens across relative distances within different *base* values, on RoPE, PSE, and mPSE.

accuracy when $\hat{m}$ is fixed at 1.5k. Essentially, reducing the value of m can achieve an effect similar to increasing the *base*. This is because a smaller m causes the onset of oscillations in PEPE to occur earlier (as shown in the middle of Fig. 4, where the oscillation onset at 4k is advanced). This early onset truncates the original declining trend of the attention scores, thereby raising the average oscillation level in PEPE.

Table 8: Passkey retrieval accuracy of PEPE methods with different RoPE *base* values.

Extension Method	*base*	Evaluation Context Length					
		2k	4k	8k	16k	32k	64k
PSE	10^4	1	1	0.6	0.35	0	0
	10^5	1	1	1	1	0.4	0
mPSE	10^4	1	1	0.35	0.3	0.05	0
	10^5	1	1	1	1	1	1

Table 9: Passkey retrieval accuracy of PEPE methods with different $\hat{m}$.

Extension Method	$\hat{m}$	Traing Steps	Evaluation Context Length					
			2k	4k	8k	16k	32k	64k
PSE	4k	100	1	1	0.6	0.4	0	0
	3k	100	1	0.8	0.8	0.1	0.1	0
	2k	100	1	1	0.6	0.5	0.2	0.1
	1.5k	100	1	1	0.8	1	0.5	0.3
	1.5k	200	1	1	1	1	0.9	0.7
mPSE	4k	100	1	1	0.1	0	0	0
	3k	100	1	1	0.7	0.4	0.1	0
	2k	100	1	0.8	0.5	0.3	0	0
	1.5k	100	1	0.9	0.9	0.9	0.9	0.6
	1.5k	250	1	0.9	1	1	1	0.9

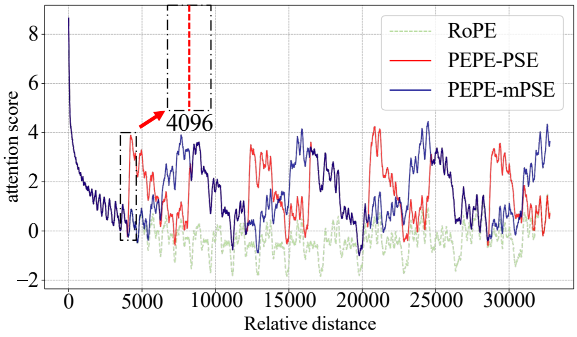


Figure 5: Attention scores of relative distances for RoPE, PSE and mPSE, evaluated on similar tokens. The application of the Savitzky-Golay filter results in smoothed attention curves, masking any apparent abrupt changes in the plot. However, at $\hat{m} = 4k$, an actual discontinuous jump in attention behavior is revealed for PSE.

A.3 Additional discussion between PSE and mPSE on attention score

We discuss PSE and mPSE from the perspective of attention scores. Fig. 5 presents the attention scores of relative distances for RoPE, PSE and mPSE, evaluated on similar tokens. We observe that PSE exhibits an abrupt change in attention scores at every $\hat{m}$-token interval, while mPSE maintains a continuous and smooth oscillation pattern. We conjecture that the smooth evolution of attention scores helps the model achieve more stable performance in passkey retrieval (as shown in the last two rows of PSE and the last two rows of mPSE in Table 9).