HoPE: A Novel Positional Encoding Without Long-Term Decay for Enhanced Context Awareness and Extrapolation

Yuhan Chen[1] **Ang Lv**[2] **Jian Luan**[1] **Bin Wang**[1] **Wei Liu**[1]*

[1]MiLM Plus, Xiaomi Inc. [2]Gaoling School of Artificial Intelligence, Renmin University of China

{chenyuhan5, luanjian, wangbin11, liuwei40}@xiaomi.com

anglv@ruc.edu.cn

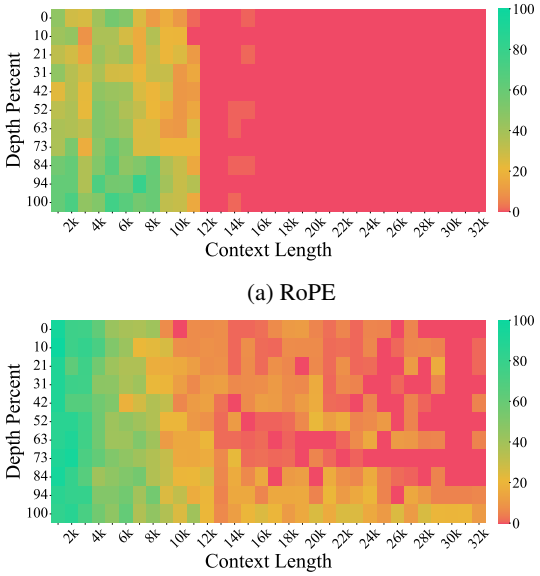
Abstract

Many positional encodings (PEs) are designed to exhibit long-term decay, based on an entrenched and long-standing inductive opinion: tokens farther away from the current position carry less relevant information. We argue that long-term decay is outdated in the era of LLMs, as LLMs are now applied to tasks demanding precise retrieval of in-context information from arbitrary positions. Firstly, we present empirical analyses on various PEs, demonstrating that models inherently learn attention with only a local-decay pattern while forming a U-shape pattern globally, contradicting the principle of long-term decay. Furthermore, we conduct a detailed analysis of rotary position encoding (RoPE, a prevalent relative positional encoding in LLMs), and found that the U-shape attention is caused by some learned components, which are also the key factor limiting RoPE's expressiveness and extrapolation. Inspired by these insights, we propose **H**igh-frequency r**o**tary **P**osition **E**ncoding (*HoPE*). *HoPE* replaces the specific components in RoPE with position-independent ones, retaining only high-frequency signals, which also breaks the principle of long-term decay in theory. *HoPE* achieves two major advantages: (1) Without constraints imposed by long-term decay, contradictory factors that limit attention optimization are removed. Thus, the model's context awareness is enhanced. (2) *HoPE* exhibits greater robustness to the out-of-distribution behavior in attention patterns during extrapolation. The effectiveness of *HoPE* is validated through extensive experiments and with a large language model of up to *3 billion* parameters.

1 Introduction

Positional encoding (PE) plays a crucial role in Transformers (Vaswani et al., 2017) to capture the order of input sequence, as the attention mechanism is permutation invariant. The original PE proposed by Vaswani et al. (2017) struggles to generalize beyond the training sequence length. To address this limitation, relative positional encoding (RPE) methods have been introduced, including RoPE (Su et al., 2021), ALiBi (Press et al., 2021), and KERPLE (Chi et al., 2022a). These RPEs share a long-standing and entrenched design (Su et al., 2021): the long-term decay, i.e., tokens with a long relative distance should receive less attention.

However, in the era of LLMs, a question arises: is it still necessary to retain this design? As LLMs are increasingly used in long-text scenar-



(a) RoPE

(b) *HoPE*

Figure 1: Performance comparison in the "Needle-in-Haystack" task between RoPE and *HoPE* using a 3B Llama-based model trained with a sequence length of 8192 over 500 billion tokens. *HoPE* demonstrates superior performance in both context awareness (0-8k) and extrapolation (8k-32k).

* Corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 23044–23056

ios—where they must leverage distant information, such as in Retrieval-Augmented Generation (RAG, Izacard and Grave, 2020)—long-term decay potentially limits their performance.

In this paper, we demonstrate that the answer to the above question is no. Through empirical analyses on various PEs, we found that the attention patterns learned by models tend to contradict the principle of long-term decay. Specifically, models only retain a local-decay pattern, while learning a U-shape attention distribution globally. We further delve into an analysis of RoPE (Rotary Position Encoding by Su et al., 2021, a widely-used RPE), which claims to ensure the long-term decay by combining various frequency components (See Section 2 for details) while empirically learns the U-shape pattern. We decomposed these components and obtained the following observations:

(1) In RoPE, certain frequency components, which we refer to as "activated" components, play a key role in shaping the final U-shaped attention pattern. Their frequencies can be pre-determined based on the training context length. Interestingly, these components exhibit fluctuations similar to the overall attention pattern. Our observations suggest that these components initially exert a predominant influence on attention patterns during early training stages. However, as training progresses, the model attempts to counterbalance their effects by increasing the weight of other components. We hypothesize that this behavior reflects a form of shortcut learning (Geirhos et al., 2020; Robinson et al., 2021; Du et al., 2022), which may impede effective optimization.

(2) We explored the attention patterns in extrapolation tasks and found that "activated" components are a key factor limiting RoPE's extrapolation abilities. These components cause out-of-distribution (OOD) attention logits in the first layer during extrapolation, triggering a cascade of disrupted attention patterns through the subsequent layers.

(3) The top low-frequency components (whose frequency is lower than the "activated" components) tend to stabilize as constant patterns, with a small magnitude. This indicates that these components are not being effectively utilized for representing positional information and learn more about semantics information.

Based on the findings above, we summarize three key insights: (1) Global long-term decay is not necessary for the model and may even hinder optimal learning. (2) To enhance the model's context awareness and extrapolation, the frequencies of RoPE's learned components should be constrained. (3) There is redundancy in RoPE, and representation subspaces occupied by certain components could be better utilized.

In this paper, we propose a novel positional encoding method called **H**igh-frequency r**o**tary **P**osition **E**ncoding (*HoPE*). *HoPE* follows above insights and is quite intuitive to implement: we replace the "doomed-to-be-activated" and top low-frequency components in the original RoPE with position-independent ones, while retaining the high-frequency components. As a result, contradictory factors for attention optimization are eliminated, extrapolation limitations are reduced, and position information is still well-represented by high-frequency signals.

On small language models with 125 million parameters, we assess the model's potential both within and beyond the context length by evaluating perplexity, in-context copying ability and few-shot following ability. *HoPE* demonstrates superior performance compared to other PEs. We further trained large language models with 3 billion parameters from scratch, and found *HoPE* performs better than RoPE in complex NLP tasks.

To sum up, we make three major contributions:

(1) We show that long-term decay in PEs is unnecessary in the era of large models, as supported by empirical analysis of various PEs.

(2) We explore the relationship between the overall attention pattern and RoPE's decomposed components, proposing a new explanation for RoPE's limited performance and poor extrapolation.

(3) Based on the above insights, we design *HoPE*, a novel relative positional encoding. Experiments empirically validate the effectiveness of *HoPE*.

2 Related Work

Positional encoding is a fundamental component of Transformer models (Vaswani et al., 2017), addressing the lack of sequential information inherent in self-attention mechanisms. While early approaches primarily relied on absolute positional encoding (APE), recent research has increasingly focused on enhancing self-attention with relative positional encoding (RPE) (Shaw et al., 2018; Raffel et al., 2019), which provides better generalization and flexibility. Currently, the popular RPE methods can be divided into two main types (Zheng et al., 2024):

rotary position encoding and additive position encoding.

Rotary position encoding (RoPE, Su et al., 2021) encodes positional information by rotating the query and key vectors. In each Transformer layer, RoPE applies a d-dimensional rotation matrix (denoted as $R_{\theta,m}$) to the query or key vector at position m in the sequence for positional encoding. The specific inner product process can be illustrated as follows:

$$\begin{aligned} q_m &= R_{\Theta,m} W_q x_m = R_{\Theta,m} q, \\ k_n &= R_{\Theta,n} W_k x_n = R_{\Theta,n} k, \\ q_m \cdot k_n &= (R_{\Theta,m} q)^\top (R_{\Theta,n} k) = q^\top R_{\Theta,m-n} k \end{aligned} \tag{1}$$

where x is the d-dimensional input of the current Transformer layer, and the matrix $R_{\Theta,m}$ is a block diagonal matrix consisting of $d/2$ blocks, each of which size 2×2 and assigned a specific angle θ. This is defined as:

$$\begin{aligned} R_{\theta_i,m} &= \begin{bmatrix} \cos(m\theta_i) & -\sin(m\theta_i) \\ \sin(m\theta_i) & \cos(m\theta_i) \end{bmatrix}, \\ R_{\Theta,m} &= \texttt{Diag}(R_{\theta_0,m}, ..., R_{\theta_{d/2-1},m}) \end{aligned} \tag{2}$$

where $\theta_i = b^{-\frac{2i}{d}}$, and b is referred to as the base of the rotary angle.

This encoding method cleverly computes the inner product of relative positions by encoding absolute positions without altering the attention computation process, making it more compatible with various efficient inference methods. However, the original RoPE encoding exhibits poor extrapolation capability for longer sequences (Press et al., 2021; Kazemnejad et al., 2023). This raises one popular research direction for exploring RoPE-based length extrapolation methods, such as PI (Chen et al., 2023), LongRoPE (Ding et al., 2024), Randomized RoPE (Ruoss et al., 2023) and YaRN (Peng et al., 2023).

Additive relative positional encoding (ARPE) is another popular method, which introduces a bias matrix B to the original (pre-softmax) attention logits. This approach can be uniformly formula as follows.

$$Attn_{ARPE}(X) = XW_Q(XW_K)^T + B \tag{3}$$

Different designs of the bias matrix B result in various APE variants, including T5's Bias (Raffel et al., 2019), ALiBi (Press et al., 2021), KERPLE (Chi et al., 2022a), Sandwich (Chi et al., 2022b), and FIRE (Li et al., 2024). These ARPE methods claim robust performance in length extrapolation, as measured by the perplexity (PPL). Nevertheless, some studies (Press et al., 2021) noted that PPL may not accurately represent real task performance. Our study further confirms that some ARPEs fail to effectively leverage global information, resulting in only marginal improvements in actual length extrapolation.

3 Discussion on Position-related Attention Pattern

In this section, We first present the position-related attention patterns (within the training context length) learned by three PEs. We observed that, although the long-term decay of PEs is intuitive, this decay is not global in the empirical attention patterns. Instead, the attention patterns tend to resemble a U-shape curve.

Secondly, we delve into a detailed analysis of the relationship between this U-shape pattern and the various components (assigned with different frequencies) of RoPE. We found that the overall pattern is strongly correlated with some components with specific frequencies, which are key factors to limit model's context awareness and extrapolation.

3.1 Experiments Setups

We train small Llama language models (Touvron et al., 2023a,b; Dubey et al., 2024) with 125 million parameters, using different PEs. The training dataset contains 200 billion tokens sourced from RedPajama (Weber et al., 2024). The training context length is 512 tokens and the update steps are 50,000. Detailed configurations and other hyperparameters are provided in the Appendix A.

To observe the position-related attention patterns both within and beyond the training context length, we set two test lengths: 512 and 1024. For each test length, we generate 5,000 corresponding data samples, each assigning a random token from the vocabulary to all positions in the input sequence (except for the initial [bos]). We then calculate the pre-softmax attention logits for each position. To illustrate a common pattern, we average the results across all layers and heads, as most heads demonstrate similar behaviors.

3.2 Long-term Decay in Attention Patterns

Our analysis focused on three PEs including learnable APE, RoPE, and KERPLE. We don't take

ALiBi into account, as its bias matrix B is unlearnable and forces the attention pattern to be global long-term decay.

Results are shown in Figure 2. One important observation is that the attention patterns **do not exhibit global long-term decay**. Instead, the attention patterns tend to form a U-shape curve, which ensures the decay of adjacent tokens while increasing the importance of the initial tokens.

3.3 Effects of Different Components of RoPE in Attention Pattern

Figure 2 also demonstrated that RoPE empirically learns the U-shape pattern while claiming to employ multiple components with different frequencies to ensure long-term decay attention. We wonder which components truly matter in this process and delve into a detailed analysis.

3.3.1 Preliminaries

Components of RoPE According to Eq.2, we can see that the dot product in attention can be broken down into an inner product process of d/2 components, each with a distinct angle θ_i, followed by a summation. This can be expressed by the following formula, which allows us to explore the individual effect of each positional component C_i.

$$
\begin{aligned}
q_m \cdot k_n = q^T R_{\Theta,m-n} k &= \sum_{i=0}^{d/2-1} \underbrace{q_i^T R_{\theta_i,m-n} k_i}_{C_i} \\
&= \sum_{i=0}^{d/2-1} ((q_{i,0}k_{i,0} + q_{i,1}k_{i,1}) \cdot cos((m-n)\theta_i)) \\
&+ (q_{i,0}k_{i,1} - q_{i,1}k_{i,0}) \cdot sin((m-n)\theta_i))
\end{aligned} \tag{4}
$$

Variance Accounted For (VAF) VAF (Yoon et al., 2021; Qiu et al., 2021) is primarily used to measure the explanatory power of components for the total variability. It serves as a crucial criterion for identifying effective principal components. A larger value indicates that the component holds greater importance. The formula is as follows:

$$
VAF_{\hat{y},y}(\%) = [1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{\sum_{i=1}^{n} y_i}] \times 100 \tag{5}
$$

where $\hat{y}$ is a component of y.

3.3.2 Experiments and Results

We decompose RoPE into several components, each associated with a unique frequency θ, and examine their individual contribution to the overall attention logits, both within the training length and during extrapolation. Additionally, under the scenarios within training context length, we tracked the pattern dynamics of components as training progresses, using VAF metric.

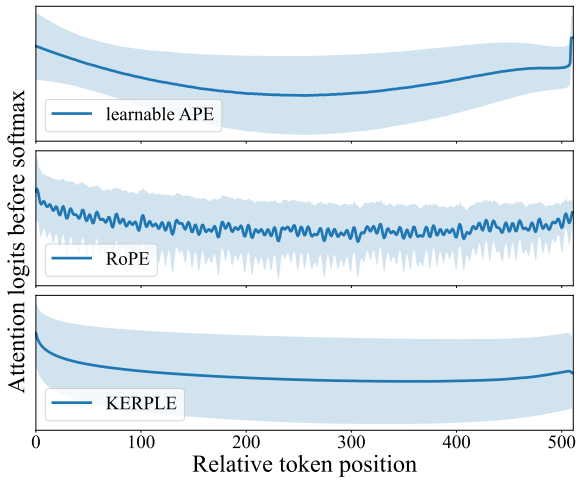
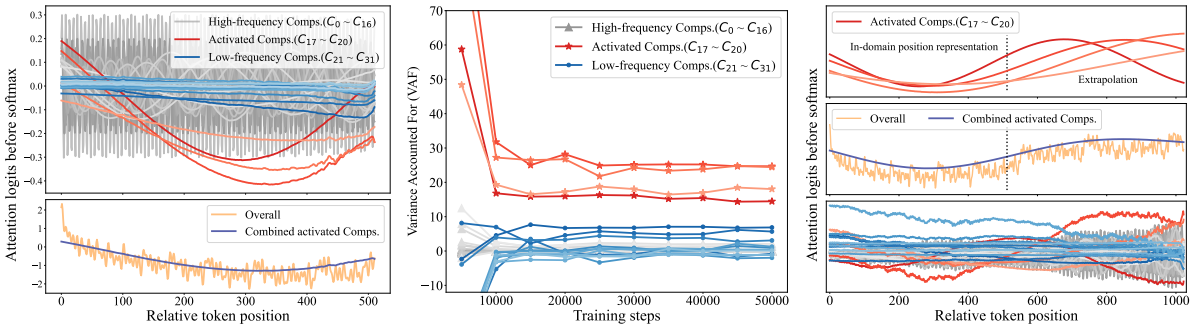


Figure 2: Attention patterns based on different learnable position encodings: (a) learnable APE, (b) RoPE, (c) KERPLE.

The results are presented in Figure 3.[1] From these results, we can derive three key insights.

(1) The learning of attention patterns is closely associated with some specific components in RoPE, while the model tends to counteract these "activated" components during training. As seen in Figure 3b, some components (referred to as "activated" components, highlighted with red in Figure 3b) exhibit high VAF, indicating that they dominate the formation of the overall U-shape pattern. The lower subplot in Figure 3a further confirms this, as the combined pattern of these components mirrors the fluctuations of the overall pattern. However, as indicated in Figure 3b, the VAF values of the "activated" components decrease as training progresses, suggesting that the model is reducing the contribution of these components. We consider this phenomenon a form of shortcut learning (Geirhos et al., 2020; Robinson et al., 2021; Du et al., 2022), which may constrain the model's overall learning. We also found that all these "activated" components exhibit U-shaped fluctuations across varying training lengths (as seen in the upper subplot of Figure 3a). Upon further examination of these components, we found that their frequencies

[1]To further validate our findings, we also included results with a longer training context length of 1024 in Appendix B.



(a) (b) (c)

Figure 3: (a) We decomposed RoPE into components (Comps.) for analysis (See Eq.4). The upper subplot displays the contribution to the overall attention logits from each component. We highlight some components with outstanding patterns, namely "activated" components, in red and lower frequency components in blue. The lower subplot presents the overall attention logits, along with the combined effects of "activated" components we highlighted in the top figure. (b) Variance Accounted For (VAF, See Section 3.3.1) for different components of RoPE during training. (c) The OOD phenomenon in extrapolation caused by "activated" components. The two upper subplots show the attention patterns of the first layer, and the lower subplot presents the anomalous patterns of the subsequent layers. The model training length presented here is 512, results for training length in 1024 can be found in Appendix B.

(θs) fall within the range $\left(\frac{\pi}{L}, \frac{2\pi}{L}\right)$, where L is the training length.

(2) The "activated" components in RoPE are also the primary reason for its poor extrapolation ability. As mentioned above, the attention pattern is closely tied to the "activated" components, which exhibit U-shape (or low half-cycle) patterns with the training length. Considering the cosine properties of these components (see Section 3.3.1 for detail), we can clearly observe from the attention pattern in the first layer (shown in Figure 3c) that these "activated" components are located in the upper half-cycle when extrapolation, which is a significant out-of-distribution (OOD) phenomenon, and subsequently leads to the disarray of attention patterns in later layers.

(3) Components with a lower frequency than the "activated" ones tend to learn a constant pattern and are not effectively utilized. Another observation is that many components exhibit a constant pattern despite their cosine properties, as shown in the upper subplot of Figure 3a. Upon delving deeper into these components, we found that their frequencies are all lower than the "activated" components. We speculate that these top low-frequency components do not represent positional information, but rather semantic information. And the properties constraints on them may even hinder this learning, as the corresponding patterns exhibit small magnitudes.

4 A Novel PE Enhances Model's Context Awareness and Exploration

Inspired by all experimental results and observations above, we proposed **H**igh-frequency r**o**tary **P**osition **E**ncoding (*HoPE*). With slight modification in RoPE, *HoPE* greatly improves the model's context awareness and extrapolation. We first detail our approach and then validate its effectiveness on perplexity, copying task, and few-shot following tasks. The results demonstrate that *HoPE* exhibits superior performance compared to other PEs.

4.1 Method

We propose our method based on the following considerations: (1) Global decay is unnecessary, thus some components in position encoding could be removed. (2) Components with U-shape fluctuations within the training length lead to shortcut learning and poor extrapolation. (3) Components with lower frequencies tend to learn semantics but are not well learned. Since both types of components belong to the low-frequency and are mostly controlled by the latter part of the $R_{\Theta,m}$ matrix in the original RoPE, we implement our approach by replacing these components with position-independent ones while retaining the high-frequency components. We call our method **H**igh-frequency r**o**tary **P**osition **E**ncoding (*HoPE*).

We first identify the "doomed-to-be-activated" components and top low-frequency components in the original RoPE. As mentioned in Section 3.3,

the frequencies (Θ_{al}) and the minimum index a of these two components could be calculated based on the training context length L. The process is as follows:

$$\Theta_{al} = \{\theta | \theta < \frac{2\pi}{L}\}, \theta \in \Theta$$
$$a = \texttt{argmax}(\Theta_{al}) \tag{6}$$

Next, we divide the query (or key) into two parts based on the index a, applying positional encoding only to the first part. For the $R_{\Theta_h,m}$ matrix applied in positional encoding, we obtain it by setting $\Theta_h = \Theta - \Theta_{al}$. The entire process is shown in the following formula.

$$\begin{aligned}
&R_{\Theta_h,m} = \texttt{Diag}(R_{\theta_0,m}, \ldots, R_{\theta_{a-1},m}) \\
&q_m = \begin{bmatrix} q_{m,h} \\ q_{m,l} \end{bmatrix}, \quad q_{m,h} = R_{\Theta_h,m} q_h, \quad q_{m,l} = q_l \\
&k_n = \begin{bmatrix} k_{n,h} \\ k_{n,l} \end{bmatrix}, \quad k_{n,h} = R_{\Theta_h,n} k_h, \quad k_{n,l} = k_l \\
&q_m \cdot k_n = q_h^\top R_{\Theta_h,m-n} k_h + q_l^\top k_l
\end{aligned} \tag{7}$$

4.2 Effect Verification of *HoPE*

4.2.1 Evaluation

Traditional methods for testing extrapolation usually use perplexity (PPL) as a metric. However, previous studies (Press et al., 2021) indicate that perplexity (PPL) does not effectively reflect a model's ability to fully leverage the context. In fact, a model can achieve lower PPL by primarily focusing on nearby tokens within the training length. Therefore, to more comprehensively assess the model's extrapolation, along with its contextual awareness and instruction-following potential, we additionally design two simple tasks: copying and few-shot learning.

Perplexity Perplexity (PPL) is a commonly used metric for evaluating a model's extrapolation capability. We conduct our evaluation on a subset of the C4 dataset (Raffel et al., 2019) with 1,000 samples by comparing the zero-shot perplexity of the last 256 tokens across different input lengths.

In-Context Copying The copying capability is one of the most fundamental abilities of language models and is closely related to token order. Many previous works (Liu et al., 2023; Golovneva et al., 2024; Lv et al., 2024) on model structure optimization have designed similar tasks to evaluate the models' effectiveness. Based on these studies, We designed our copying task. Specifically, we constructed a test set containing 500 samples, with each sample consisting of multiple sequences. Each sequence has an average length of 12 tokens, with a unique 8-gram prefix and 4-gram suffix. During testing, we concatenate a specific number of sequences with the prefix of a certain i-th sequence (queried sequence) to serve as the model's input. The model's objective is to output the suffix of the queried prefix, with the middle sequence selected as the queried one. Figure 4 illustrates a specific example input case.

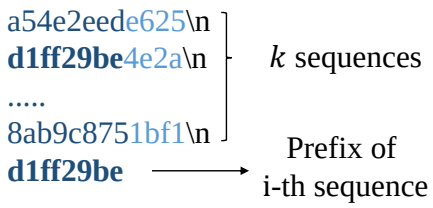


Figure 4: Specific input example of copying task.

Few-shot Following Few-shot learning is another core ability of the model and serves as the foundation for instruction following. We created a test set with 600 samples selected from three tasks (SST-2, QNLI and RTE) in the GLUE (Wang et al., 2018) benchmark. For each input, we concatenate few-shot examples, a set of meaningless sentences, and the queried input. Specific examples can be found in Figure 9. As for the evaluation metric, instead of focusing on actual accuracy, we emphasize whether the model's output falls into the label sets from the few-shot examples. For instance, if the label set in the contextual examples is $0, 1$, the model's output, whether 0 or 1, will be counted. And we define this as a measure of follow ability (FA). We present the average performance across the three tasks in the main text, and detailed results for each task please refer to Appendix C.2.

We set the rotary base $b = 10,000$ in *HoPE*. Other settings are consistent with those in Section 3.1. We evaluate the proposed *HoPE* against a range of established baselines, including RoPE (Su et al., 2021), ALiBi (Press et al., 2021), KERPLE (Chi et al., 2022a), and FIRE (Li et al., 2024), as well as two typical RoPE-based extrapolation methods: PI (Chen et al., 2023) and YaRN (Peng et al., 2023).

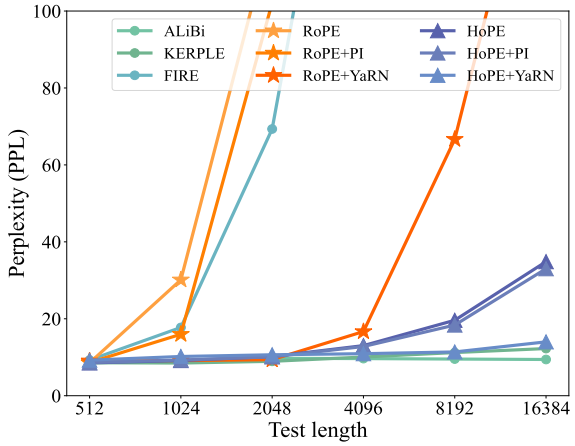


Figure 5: Perplexity Comparison on C4 dataset.

Method	Sequence Number						
	30	40	50	60	70	80	Avg.
ALiBi	78.00	64.00	25.90	8.30	2.30	1.20	29.95
KERPLE	80.20	74.00	64.60	56.80	25.60	21.70	53.82
FIRE	70.00	43.20	28.20	6.40	2.80	0.20	25.13
RoPE	77.60	55.80	9.40	0.00	0.00	0.00	23.80
+PI	76.40	61.40	20.20	4.60	0.00	0.00	27.10
+YaRN	65.20	49.80	64.40	50.80	48.60	39.40	54.20
Our *HoPE*	**84.00**	**77.00**	**77.00**	60.40	32.40	30.60	60.23
+PI	80.20	74.20	73.60	60.60	36.80	39.40	60.80
+YaRN	78.60	73.20	76.00	**69.00**	**68.40**	**46.60**	**68.63**

Table 1: Results of the copy task. We highlight the leading results with **bold fonts**. The input length is under the training length of 512 when the sequence number is below 60.

4.2.2 Results

The results for perplexity (PPL), copying task, and few-shot following task are presented in Figure 5, Table 1 and Table 2, respectively. From these results, we can draw the following conclusions:

(1) **From all perspectives in the figure and tables above, it can be confirmed that our approach significantly enhances the context awareness and extrapolation of the original RoPE.** As shown in Figure 5, our method noticeably smooths the increase in PPL observed in RoPE, achieving low PPL even with training lengths 4 times longer or more. It records a PPL of 8.5241 at 512, and 13.0257 at 4,096. Table 1 and 2 further demonstrate that our approach not only improves extrapolation but also enhances context awareness within the training length. Specifically, compared to RoPE, the model's copy ability increased from an average of 23.80 to 60.23, while its few-shot following capability improved from an average of 54.10 to 79.20.

(2) **When combined with extrapolation methods like PI and YaRN, our method achieves even better extrapolation results.** As shown in Table 1 and 2, our method, when integrated with YaRN, achieves the best overall performance across all PEs, with an average score of 68.63 in the copy task and 93.77 in the few-shot following task. However, as also noted in the tables, while both PI and YaRN enhance extrapolation, they appear to negatively impact the model's context awareness within the training length.

Method	Input Lengths					
	256	512	768	1024	1280	Avg.
ALiBi	99.67	85.67	68.67	6.33	1.00	52.27
KERPLE	77.17	70.17	22.33	14.67	16.33	40.13
FIRE	87.17	86.33	32.17	19.50	19.33	48.90
RoPE	98.17	97.00	51.33	17.00	7.00	54.10
+*PI*	98.17	**98.83**	66.67	38.50	8.00	62.03
+*YaRN*	**99.83**	85.67	91.00	81.67	20.33	75.70
Our *HoPE*	99.67	98.33	74.50	67.83	55.67	79.20
+*PI*	99.17	98.50	83.67	78.67	51.67	82.33
+*YaRN*	97.33	92.17	**91.67**	**88.67**	**99.00**	**93.77**

Table 2: The results of the few-shot following experiment. We measure the model's following ability (FA), which counts the instances when the output includes one of the label sets from the examples. The leading results are highlighted with **bold fonts**.

(3) **Relying solely on perplexity (PPL) to measure extrapolation is not reliable.** In some cases, the PPL measurements (shown in Figure 5) contradict the performance in other tasks (shown in Table 1 and 2), indicating that PPL may not prove a method's ability to effectively utilize global information. It might reflect "pseudo" extrapolation, as seen in the result of ALiBi and KERPLE. From Table 1 and 2, it is evident that ALiBi's actual extrapolation is poor, and while KERPLE shows some extrapolation, it is not as strong as the PPL suggests and performs slightly worse in few-shot following performance.

4.3 Attention Patterns in *HoPE*

To better understand how *HoPE* functions, we present its learned position-related attention pattern, as depicted in Figure 6. As shown in the upper subplot, *HoPE* demonstrates a U-shaped fluctuation similar to RoPE within the training length. The positional fluctuation in *HoPE* is milder, suggesting better adaptation to long-context tasks where semantic information is more critical. In terms of extrapolation patterns (as shown in the lower subplot of Figure 6), *HoPE* appears not to exhibit the out-of-distribution (OOD) behavior observed

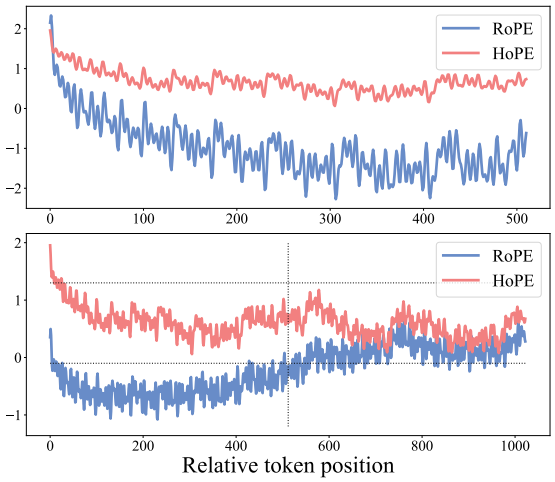


Figure 6: Attention patterns in *HoPE* and RoPE, both within (top) and beyond (bottom) training length.

in RoPE, which could explain its superior extrapolation capabilities.

5 Ablation Study of *HoPE*

In this section, we conducted an ablation study of our method and validated its effectiveness through measurements on the copy task.

Settings We mainly performed three ablation settings: (1) **AB1**, replacing only the "activated" components with positional-independent components. (2) **AB2**, replacing only the top low-frequency components with positional-independent components. (3) **AB3**, removing both components without adding positional-independent ones. We do this by replacing these components with high-frequency components.

Results As shown in Table 3, removing the "activated" components (AB1) results in a significant improvement in both context awareness and extrapolation, with an average increase of 14.5 points over RoPE. This outcome suggests that these components indeed contribute to the model's shortcut learning, hindering optimal learning.

Removing the top low-frequency components (AB2) helps improve the model's context awareness but contributes less to extrapolation. This confirms that the "activated" components are the key factor behind poor extrapolation performance.

Additionally, we can observe that AB3 shows a significant decline in both context awareness and extrapolation, highlighting the importance of the position-independent components. This indicates that the model indeed requires certain components

Method	Sequence Number						
	30	40	50	60	70	80	Avg.
RoPE	77.60	55.80	9.40	0.00	0.00	0.00	23.80
AB1	83.80	68.80	17.80	21.80	12.00	25.60	38.30
AB2	81.40	59.60	16.80	1.00	0.00	0.00	24.47
AB3	65.60	30.6	6.20	2.00	0.80	0.00	17.53
HoPE	**84.00**	**77.00**	**77.00**	**60.40**	**32.40**	**30.60**	**60.23**

Table 3: The results of the ablation study. AB1 means only removing the "activated" components, AB2 means removing the top low-frequency components, and AB3 refers to removing both types of components but without the position-independent components.

to learn semantic information. The slight improvement (an average increase of 0.67 points) from AB2 further suggests that the original low-frequency components in RoPE effectively fulfill this role, while they have not been fully learned.

Based on the results above, we have demonstrated the rationale behind our *HoPE*'s design and identified the source of its performance improvements.

6 Scalability of *HoPE*

Based on our understanding of *HoPE*'s advantages, derived from a series of empirical experiments with small models and toy tasks, we conducted further comparative experiments on real-world tasks. These experiments involved training large language models from scratch, comparing RoPE with *HoPE*.

Settings We trained a Llama-based model with 3 billion parameters. The training was conducted over 120,000 steps with a sequence length of 8192, using approximately 500 billion tokens in total. Appendix A provides the model configurations and training details. For evaluation, we selected eight general-purpose benchmarks to assess **context awareness**, including MMLU (5-shot) (Hendrycks et al., 2020), MMLU-PRO (5-shot) (Wang et al., 2024), GPQA (0-shot) (Rein et al., 2023), BBH (3-shot) (Srivastava et al., 2023), WinoGrande (5-shot) (Sakaguchi et al., 2019), GSM8k (8-shot) (Cobbe et al., 2021), MATH (4-shot) (Lightman et al., 2023), and DROP (3-shot) (Dua et al., 2019). The shot count settings follow the standard configurations used in prior works (Dubey et al., 2024; Bai et al., 2023a). For **extrapolation**, we employ the LongBench (Bai et al., 2023b), which comprises six task categories: single-document QA, multi-document QA, summarization, few-shot reasoning, code completion, and synthetic tasks. We also conduct Needle-in-a-

Benchmark	MMLU	MMLU-PRO	GPQA	BBH	WinoGrande	GSM8k	MATH	DROP	AVG.
RoPE	34.27	12.60	23.23	29.00	**51.70**	10.61	1.16	31.29	24.23
HoPE (Ours)	**38.38**	**12.74**	**28.28**	**29.15**	50.43	**12.05**	**1.84**	**38.46**	**26.42**

Table 4: Performance comparison between RoPE and *HoPE* across eight benchmarks using the 3B Llama-based model. Better results are highlighted in **bold fonts**. *HoPE* demonstrates superior performance in most tasks.

Task	RoPE	*HoPE*
Single-doc QA	14.74	**17.87**
Multi-doc QA	5.22	**9.74**
Summarization	12.98	**17.63**
Few shot	23.47	**47.50**
Code	35.40	**49.47**
Synthetic	**2.66**	2.05

Table 5: Performance comparison between RoPE and *HoPE* across 6 major tasks of LongBench using the 3B Llama-based model. Better results are highlighted in **bold fonts**. *HoPE* demonstrates clear advantages in most tasks.

Haystack tests (gkamradt, 2023) to provide a comprehensive evaluation of both context awareness and extrapolation. We use OpenCompass (Fu et al., 2024) to compute the results.

Results As shown in Table 4, our *HoPE* achieves an average score of 26.42, outperforming RoPE's score of 24.23. Notable improvements are observed on various benchmarks, such as MMLU (+4.11), GPQA (+5.05), and DROP (+7.17). The validation loss curves presented in Appendix D further support these results, indicating consistently lower training loss for *HoPE*.

In terms of extrapolation, Table 5 shows that *HoPE* achieves notable gains on LongBench, including Single-doc QA (+3.13), Multi-doc QA (+4.51), Summarization (+4.66), Few-shot learning (+24.03), and Code (+14.07), demonstrating its clear advantages in long-context settings.

In addition, results on the "Needle-in-Haystack" task (depicted in Figure 1) further illustrate its significant extrapolation potential, even at a sequence length of 32k. These findings highlight *HoPE*'s potential as a strong alternative to RoPE in advancing the next generation of state-of-the-art LLMs.

7 Conclusion

In this paper, we explore the empirical attention patterns of various positional encodings and observe that position-related attention tend to form a U-shape pattern, benefiting more from local decay rather than global. Our further analysis of RoPE reveals a strong correlation between the U-shape pattern and its learned components. We identify that certain "activated" components and top low-frequency components in RoPE hinder the model's optimal learning process, limiting its context awareness and extrapolation. Consequently, we propose our method, *HoPE*, which breaks the principle of long-term decay in theory, allowing for optimal utilization of components for positional encoding. Extensive experiments demonstrate its effectiveness in enhancing both context awareness and extrapolation.

8 Limitations

In this paper, we introduce a novel and effective positional encoding method to improve the model's context awareness and extrapolation capabilities. However, due to limited resources, we have only implemented our method using the vanilla attention mechanism. We recognize that there are various variants of attention mechanisms and see great potential in exploring our method on them in future studies.

The potential risks associated with our research align with those of other endeavors involving large language models, including misuse for generating harmful content, perpetuation of biases, data privacy concerns, and environmental costs linked to computational resource consumption.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenhang Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, K. Lu, and 31 others. 2023a. Qwen technical report. *ArXiv*, abs/2309.16609.

Yushi Bai, Xin Lv, Jiajie Zhang, Hong Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023b. Longbench: A bilingual, multitask benchmark for long context understanding. *ArXiv*, abs/2308.14508.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending context window of large language models via positional interpolation. *ArXiv*, abs/2306.15595.

Ta-Chung Chi, Ting-Han Fan, Peter J. Ramadge, and Alexander I. Rudnicky. 2022a. Kerple: Kernelized relative positional embedding for length extrapolation. *ArXiv*, abs/2205.09921.

Ta-Chung Chi, Ting-Han Fan, Alex Rudnicky, and Peter J. Ramadge. 2022b. Dissecting transformer length extrapolation via the lens of receptive field analysis. In *Annual Meeting of the Association for Computational Linguistics*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168.

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*, abs/2402.13753.

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. 2022. Shortcut learning of large language models in natural language understanding. *Communications of the ACM*, 67:110 – 120.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *North American Chapter of the Association for Computational Linguistics*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony S. Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 510 others. 2024. The llama 3 herd of models. *ArXiv*, abs/2407.21783.

Xi Fu, Shentong Mo, Anqi Shao, Anouchka P. Laurent, Alejandro L Buendia, Adolfo A. Ferrando, Alberto Ciccia, Yanyan Lan, Teresa Palomero, David M. Owens, Eric P. Xing, and Raúl Rabadán. 2024. Get: a foundation model of transcription across human cell types. *bioRxiv*.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2:665 – 673.

gkamradt. 2023. Llmtest needle in a haystack - pressure testing llms. `https://github.com/gkamradt/LLMTest_NeedleInAHaystack`.

Olga Golovneva, Tianlu Wang, Jason Weston, and Sainbayar Sukhbaatar. 2024. Contextual position encoding: Learning to count what's important. *ArXiv*, abs/2405.18719.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Xiaodong Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *ArXiv*, abs/2009.03300.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *ArXiv*, abs/2007.01282.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. 2023. The impact of positional encoding on length generalization in transformers. *ArXiv*, abs/2305.19466.

Shanda Li, Chong You, Guru Guruganesh, Joshua Ainslie, Santiago Ontanon, Manzil Zaheer, Sumit K. Sanghai, Yiming Yang, Sanjiv Kumar, and Srinadh Bhojanapalli. 2024. Functional interpolation for relative positions improves long context transformers. *ArXiv*, abs/2310.04418.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *ArXiv*, abs/2305.20050.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Ang Lv, Ruobing Xie, Xingwu Sun, Zhanhui Kang, and Rui Yan. 2024. Language models "grok" to copy. In *North American Chapter of the Association for Computational Linguistics*.

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

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

Yingui Qiu, Jian Zhou, Manoj Khandelwal, Haitao Yang, Peixia Yang, and Chuanqi Li. 2021. Performance evaluation of hybrid woa-xgboost, gwo-xgboost and bo-xgboost models to predict blast-induced ground vibration. *Engineering with Computers*, 38:4145 – 4162.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *ArXiv*, abs/2311.12022.

Joshua Robinson, Li Sun, Ke Yu, K. Batmanghelich, Stefanie Jegelka, and Suvrit Sra. 2021. Can contrastive learning avoid shortcut solutions? *Advances in neural information processing systems*, 34:4974–4986.

Anian Ruoss, Gr'egoire Del'etang, Tim Genewein, Jordi Grau-Moya, R. Csordás, Mehdi Abbana Bennani, Shane Legg, and Joel Veness. 2023. Randomized positional encodings boost length generalization of transformers. *ArXiv*, abs/2305.16843.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande. *Communications of the ACM*, 64:99 – 106.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *North American Chapter of the Association for Computational Linguistics*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, and 431 others. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Trans. Mach. Learn. Res.*, 2023.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Neural Information Processing Systems*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *BlackboxNLP@EMNLP*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max W.F. Ku, Kai Wang, Alex Zhuang, Rongqi "Richard" Fan, Xiang Yue, and Wenhu Chen. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *ArXiv*, abs/2406.01574.

Maurice Weber, Dan Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher R'e, Irina Rish, and Ce Zhang. 2024. Redpajama: an open dataset for training large language models. *ArXiv*, abs/2411.12372.

Kyubaek Yoon, Hojun You, Wei-Ying Wu, Chae Young Lim, Jongeun Choi, Connor J. Boss, Ahmed Ramadan, John M. Popovich, Jacek Cholewicki, N. Peter Reeves, and Clark J. Radcliffe. 2021. Regularized nonlinear regression for simultaneously selecting and estimating key model parameters. *ArXiv*, abs/2104.11426.

Chuanyang Zheng, Yihang Gao, Han Shi, Minbin Huang, Jingyao Li, Jing Xiong, Xiaozhe Ren, Michael Ng, Xin Jiang, Zhenguo Li, and Yu Li. 2024. Dape: Data-adaptive positional encoding for length extrapolation. In *Neural Information Processing Systems*.

A Model Configuration

Detail settings across model sizes are depicted in Table 6. All experiments use Llama tokenizer with a vocabulary of 32,000 tokens. Other hyperparameters are as follows: the AdamW optimizer is used with $(\beta_1, \beta_2) = (0.9, 0.999)$, a learning rate of $3e^{-4}$, $2,000$ warm-up steps, and a gradient clipping value of 1. Experiments for the 125M models are conducted on 8 A100 GPUs, while those for the 3B models use 256 A100 GPUs.

Hyperparameters	125M	3B
Training sequence length	512	8192
Batch size	64×8	2×256
Number of Iterations	50k	120K
Dropout Prob.	0.0	0.0
Number of Layers	12	34
Attention Head	12	16
Feature Dimension	768	2048
Intermediate Dimension	2688	8704
Precision	BFloat16	BFloat16

Table 6: Model configurations.

B Supplementary Results on the Exploration of Component Effects in RoPE Attention Patterns

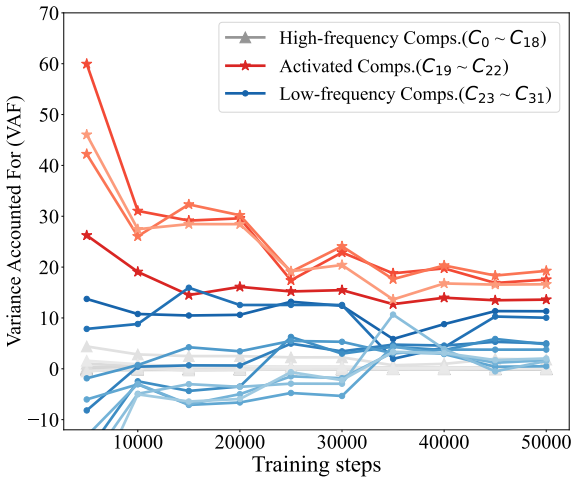


Figure 7: VAF results of each component at training length 1024.

We present supplementary experiments with training lengths of 1024, depicted in Figure 7 and 8. We reached the same conclusion as in the main text. It can be seen that "activated" and lower frequency components shift further back as the training length increases. These "activated" components still exhibit a U-shaped curve, similar to the final pattern. The lower frequency components continue to learn a constant pattern.

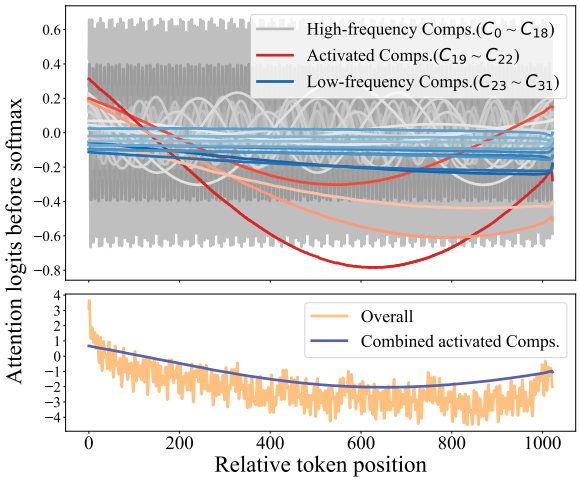


Figure 8: The attention pattern of RoPE at training length 1024.

C Supplementary for Few-shot Following Tasks

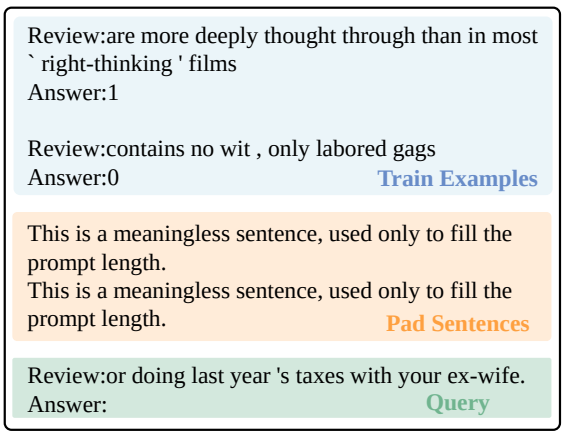


Figure 9: Input examples of few-shot following tasks. We take SST-2 as an instance.

C.1 Input Examples of Few-shot Following Tasks

Specific examples of few-shot following tasks can be found in Figure 9. In practice, we provide 5-shot training examples as context. For each instance, we dynamically pad it with a different number of meaningless sentences to ensure the various input lengths.

Method	SST-2						QNLI						RTE					
	256	512	768	1024	1280	Avg.	256	512	768	1024	1280	Avg.	256	512	768	1024	1280	Avg.
ALiBi	99.00	58.00	29.00	4.00	0.00	38.00	100.00	99.00	79.00	0.00	0.00	55.60	100.00	100.00	98.00	15.00	3.00	63.20
KERPLE	50.50	37.50	18.00	9.00	6.00	24.20	100.00	95.00	32.00	7.00	4.00	47.60	81.00	78.00	17.00	28.00	39.00	48.60
FIRE	99.50	99.00	76.50	31.50	3.00	61.90	99.00	99.00	18.00	19.00	12.00	49.40	63.00	61.00	2.00	8.00	43.00	35.40
RoPE	99.50	96.00	20.00	0.00	0.00	43.10	95.00	95.00	85.00	22.00	9.00	61.20	100.00	100.00	49.00	29.00	12.00	58.00
+ PI	99.50	96.50	33.00	4.50	0.00	46.70	100.00	100.00	94.00	75.00	11.00	76.00	95.00	100.00	73.00	36.00	13.00	63.40
+ YaRN	99.50	57.00	73.00	73.00	0.00	60.50	100.00	100.00	100.00	95.00	9.00	80.80	100.00	100.00	100.00	77.00	52.00	85.80
Our *HoPE*	99.00	95.00	45.50	52.50	32.00	64.80	100.00	100.00	100.00	88.00	37.00	85.00	100.00	100.00	78.00	63.00	98.00	87.80
+ PI	97.50	97.50	97.00	37.00	89.00	83.60	100.00	100.00	100.00	100.00	55.00	91.00	100.00	98.00	54.00	99.00	11.00	72.40
+ YaRN	100.00	97.50	76.00	69.00	98.00	88.10	92.00	80.00	99.00	99.00	100.00	94.00	100.00	99.00	100.00	98.00	99.00	99.20

Table 7: Detail results on three few-shot following tasks.

C.2 Detail Results on Few-shot Following Task

We present the average results in Table 2 of the main text. Detailed results in three tasks are depicted in Table 7.

D Supplementary Results of the 3B Llama-based Model

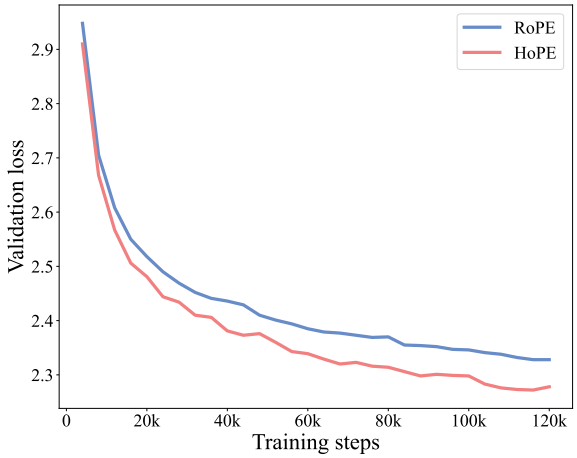


Figure 10: Comparison of validation loss between RoPE and *HoPE*using a 3B Llama-based model.

The comparison of validation loss between RoPE and *HoPE*using the 3B Llama-based Model is depicted in Figure 10.