

WAVELET-BASED POSITIONAL REPRESENTATION FOR LONG CONTEXT

Anonymous authors

Paper under double-blind review

ABSTRACT

In the realm of large-scale language models, a significant challenge arises when extrapolating sequences beyond the maximum allowable length. This is because the model’s position embedding mechanisms are limited to positions encountered during training, thus preventing effective representation of positions in longer sequences. We analyzed conventional position encoding methods for long contexts and found the following characteristics. (1) When the representation dimension is regarded as the time axis, Rotary Position Embedding (RoPE) can be interpreted as a restricted wavelet transform using simple Haar wavelets. However, because it only uses a single window size, it does not fully exploit the advantages of wavelet transforms, which capture the fine movements of non-stationary signals using multiple scales (window sizes). This limitation could explain why RoPE performs poorly in extrapolation. (2) Previous research as well as our own analysis indicates that Attention with Linear Biases (ALiBi) functions similarly to windowed attention, using windows of varying sizes. However, it has limitations in capturing deep dependencies because it restricts the receptive field of the model. From these insights, we propose a new position representation method that captures multiple scales (i.e., window sizes) by leveraging wavelet transforms without limiting the model’s attention field. Experimental results show that this new method improves the performance of the model in both short and long contexts. In particular, our method allows extrapolation of position information without limiting the model’s attention field.

1 INTRODUCTION

Several pre-trained large language models based on Transformer architecture (Vaswani et al., 2017) have demonstrated robust capabilities in various generative tasks (Devlin et al., 2019; Raffel et al., 2020; Brown et al., 2020; Touvron et al., 2023a; Jiang et al., 2023). However, limitations on the input sequence length arise due to the computational resource constraints encountered during the pre-training phase. Such constraints necessitate a determination of the maximum allowable length of sequences, hereinafter L_{train} , prior to the pre-training process, thus hindering the model’s performance in processing sequences longer than those encountered during training. This weakness is primarily attributed to the positional encoding’s ineffectiveness in handling sequences that exceed the length of those encountered during the model’s training phase (Devlin et al., 2019; Press et al., 2022).

Rotary Position Embedding (RoPE) (Su et al., 2021) has become a common approach in many language models that handle long contexts, and it employs a rotation matrix to encode positional information and facilitate the processing of long sequences. To manage sequences longer than those encountered during training, various scaling strategies (Chen et al., 2023; bloc97, 2023; Peng et al., 2024; Liu et al., 2024) have been applied to RoPE, although these often require additional fine-tuning and incur additional learning costs in addition to pre-training. In contrast, Attention with Linear Biases (ALiBi) (Press et al., 2022) is able to sequence length estimation beyond the limits of pre-training without requiring further fine-tuning. However, ALiBi limits the attention’s receptive field (Chi et al., 2023) in the manner of windowed attention (Beltagy et al., 2020). For this reason, a model using ALiBi may not be able to obtain information that is in a distant dependency relationship.

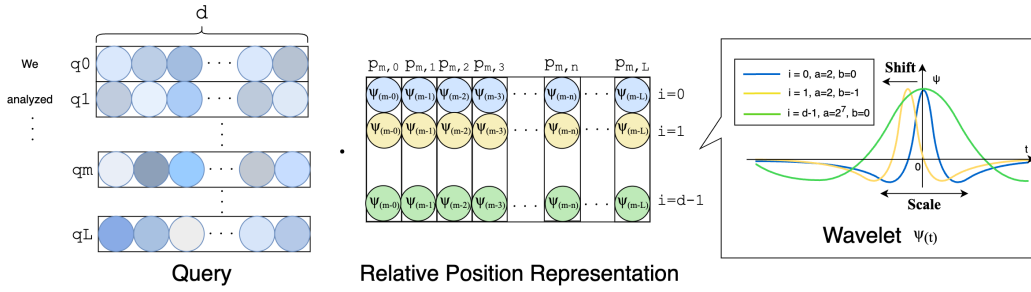


Figure 1: Overview of Wavelet-based Relative Positional Representation. As in RPE (Shaw et al., 2018), our method computes a relative positional representation $(p_{m,n})^T$ to the query q_m and the key k_n . Instead of learnable embedding in RPE, the position is computed based on the wavelet function. Different wavelet functions $\psi_{a,b}$ are used for each dimension of the head d . The scale parameter a and the shift parameter b change depending on the dimension of the head d .

In this paper, we analyze conventional positional encoding methods for long contexts, and we propose a novel positional representation that permits extrapolation without constraining the attention mechanism’s receptive field. First, we mathematically show that RoPE performs a process similar to a wavelet transformation—considered the gold standard of time-frequency analysis methodology. We interpreted the position of each token in the sequence as a time in time-frequency analysis. However, RoPE does not perform a transformation in accordance with the order of positions but rather in accordance with the number of dimensions, and it does not capture the dynamic change in a signal over time. Furthermore, the values corresponding to the wavelet scale (i.e., window size) are constant, so RoPE does not make good use of the key characteristic of wavelet transforms, which is the ability to analyze signals on multiple scales. In other words, RoPE may fail to capture the dynamic change in a signal over time, such as what occurs in natural language. In this study, we also show that ALiBi provides different window sizes for each head.

Based on these insights, we propose a wavelet transform-based method, using multiple window sizes, to offer a robust and flexible approach to positional encoding. By performing a wavelet transform along the order of positions and introducing various scale parameters, our method can capture the dynamic changes in a sequence over positions in the manner of the original feature of wavelet transformation—time-frequency analysis. Following the methodology of Relative Position Representation (RPE) (Shaw et al., 2018), we implement our method with relative ease.

From our experiments on extrapolation capabilities using the wikitext-103 dataset (Merity et al., 2017), the results demonstrate that our method surpasses traditional positional encoding methods in perplexity. We also report that our method has lower perplexity than RoPE in experiments with long contexts using the Llama-2 model (Touvron et al., 2023b) and the CodeParrot dataset.

2 BACKGROUND

2.1 POSITIONAL REPRESENTATION

Within the Transformer architecture, positional encoding is employed to accurately represent the sequential position of each token. Positional encoding can be divided into two main types: absolute position, which expresses the position of a token from the beginning of the sequence, and relative position, which expresses the position of each token within the sequence. RoPE (Su et al., 2021), a type of absolute position, uses a rotation matrix to compute the position and then multiplies it by the query and key to represent the position. RPE (Shaw et al., 2018), a type of relative position, uses a learnable embedding that represents the position of distances of up to 16 or 32 tokens by clipping. Two other examples include T5 Bias (Raffel et al., 2020), which has an enlarged RPE window size, and Transformer-XL (Dai et al., 2019), which uses a sine wave for position representation instead of learnable embedding.

Position encoding plays a critical role in enabling models to effectively handle long context sequences, and it allows for extrapolation. Relative position is not a position expression that depends

on the length of the sequence, so it is effective in extrapolation. ALiBi (Press et al., 2022) is an effective position representation method for extrapolation: It uses the relative position bias of all tokens by adding a linear bias to each head’s attention score, rather than using position embedding. However, ALiBi is unable to obtain information in a distant dependency relationship due to its constraints on the self-attention mechanism’s receptive field (Chi et al., 2023). On the other hand, absolute position is unsuitable for extrapolation because it expresses the position of all words in the sequence. For this reason, many methods have been proposed for fine-tuning RoPE using absolute position by interpolating positions (Chen et al., 2023; bloc97, 2023; Peng et al., 2024).

2.2 FREQUENCY ANALYSIS AND TIME-FREQUENCY ANALYSIS

Frequency analysis in signal processing involves analyzing the frequency components of a signal to understand its behavior. The Fourier transform (FT) (Bracewell & Bracewell, 1986) is a key method for frequency analysis, converting a signal from the time domain to the frequency domain, thus providing a global view of its frequency content. However, the FT does not provide any information about **when** specific frequencies occur. To address this limitation, time-frequency analysis techniques are applied. The wavelet transform (WT) (Grossmann & Morlet, 1984; Mallat, 1989) offers a more flexible approach by analyzing the signal at multiple scales or resolutions. The WT adaptively provides high time resolution for high-frequency components and high frequency resolution for low-frequency components, making it well-suited for analyzing signals with non-stationary or transient features. This adaptability allows the wavelet transform to capture both time and frequency information with varying degrees of precision.

3 ROPE AND WAVELET TRANSFORM

3.1 PRELIMINARY

Wavelet Transform Wavelet (wave-let) is a wave that decays quickly and locally as it approaches zero. A function ψ defined on a real \mathbb{R} is called a wavelet function if it belongs to the space $L^2(\mathbb{R})$ of square integrable functions and satisfies the following conditions:

$$\int_{-\infty}^{\infty} |\psi(x)|^2 dx < \infty. \quad (1)$$

The wavelet function is defined as follows.

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right). \quad (2)$$

In this case, b is the shift and $a > 0$ is the scale parameter. The scale parameter a simultaneously changes the range over which the wavelet is localized and the amplitude of the wavelet. Typical wavelets include the Haar wavelet (Haar, 1910), Ricker wavelet (Ricker, 1944), and Morlet wavelet (Bernardino & Santos-Victor, 2005). **Suppose that we sample T values at regular intervals from a continuous signal.** A wavelet transforms (Grossmann & Morlet, 1984) to the frequency domain and time domain by calculating the inner product of the wavelet function $\psi_{a,b}(t)$ and the signal $x(t)$:

$$W(a, b) = \sum_{t=0}^{T-1} \psi_{a,b}(t) x(t). \quad (3)$$

In some cases, the term "Discrete Wavelet Transform" or "Wavelet Transform" is used to refer to multi-resolution analysis (Mallat, 1989), but in this paper we follow the original definition. We can see that the FT only converts to the frequency domain, whereas the WT converts to two domains: scale a and shift b . For example, consider the case of converting to two scales and four shifts. When $a \in [2, 4]$ and $b \in [0, 1, 2, 3]$, the wavelet transform can be expressed in terms of determinants as follows:

$$\begin{bmatrix} W(2, 0) \\ W(4, 0) \\ W(2, 1) \\ W(4, 1) \\ \vdots \\ W(4, 3) \end{bmatrix} = \begin{bmatrix} \psi_{2,0}(0) & \psi_{2,0}(1) & \psi_{2,0}(2) & \dots & \psi_{2,0}(T-1) \\ \psi_{4,0}(0) & \psi_{4,0}(1) & \psi_{4,0}(2) & \dots & \psi_{4,0}(T-1) \\ \psi_{2,1}(0) & \psi_{2,1}(1) & \psi_{2,1}(2) & \dots & \psi_{2,1}(T-1) \\ \psi_{4,1}(0) & \psi_{4,1}(1) & \psi_{4,1}(2) & \dots & \psi_{4,1}(T-1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \psi_{4,3}(0) & \psi_{4,3}(1) & \psi_{4,3}(2) & \dots & \psi_{4,3}(T-1) \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(T-1) \end{bmatrix}. \quad (4)$$

Furthermore, since $\psi_{a,b}(t) = \psi_{a,0}(t-b)$ from Eq.2, the wavelet transform is expressed as follows.

$$\begin{bmatrix} W(2,0) \\ W(4,0) \\ W(2,1) \\ W(4,1) \\ \vdots \\ W(4,3) \end{bmatrix} = \begin{bmatrix} \psi_{2,0}(0) & \psi_{2,0}(1) & \psi_{2,0}(2) & \dots & \psi_{2,0}(T-1) \\ \psi_{4,0}(0) & \psi_{4,0}(1) & \psi_{4,0}(2) & \dots & \psi_{4,0}(T-1) \\ \psi_{2,0}(-1) & \psi_{2,0}(0) & \psi_{2,0}(1) & \dots & \psi_{2,0}(T-2) \\ \psi_{4,0}(-1) & \psi_{4,0}(0) & \psi_{4,0}(1) & \dots & \psi_{4,0}(T-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \psi_{4,0}(-3) & \psi_{4,0}(-2) & \psi_{4,0}(-1) & \dots & \psi_{4,0}(T-3) \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(T-1) \end{bmatrix}. \quad (5)$$

Due to the characteristics of the scale parameter a , the values of the wavelet matrix become 0 or approach 0 outside a certain range. This range depends on the specific wavelet function.

RoPE RoPE incorporates positional information directly into the self-attention mechanism by rotating the query and key vectors in the complex space. When divided into even and odd dimensions, the following calculations are performed for the m -th query in each sequence. In even dimensions, RoPE is expressed as follows.

$$\begin{bmatrix} q_0^m \\ q_2^m \\ \vdots \\ q_{d-2}^m \end{bmatrix} = \begin{bmatrix} \cos m\theta_1 & -\sin m\theta_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \cos m\theta_2 & -\sin m\theta_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \cos m\theta_{d/2} & -\sin m\theta_{d/2} \end{bmatrix} \begin{bmatrix} q_0^m \\ q_1^m \\ \vdots \\ q_{d-2}^m \\ q_{d-1}^m \end{bmatrix}. \quad (6)$$

where $q^m \in \mathbb{R}^{1 \times d}$ is the m -th query when the number of dimensions is d and $\theta_i = 10000^{-2(i-1)/d}$, $i \in [1, 2, \dots, d/2]$. For RoPE in odd dimensions, see Appendix A.1. The same process is also performed for the n -th key $k^n \in \mathbb{R}^{1 \times d}$.

3.2 THEORETICAL ANALYSIS

First, we show the wavelet transform using the following two Haar-like wavelets (Haar, 1910).

$$\psi(t) = \begin{cases} \cos f(t) & 0 \leq t < 1, \\ -\sin f(t) & 1 \leq t < 2, \\ 0 & \text{otherwise.} \end{cases} \quad \psi'(t) = \begin{cases} \sin f(t) & 0 \leq t < 1, \\ \cos f(t) & 1 \leq t < 2, \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

where $f: \mathbb{R} \rightarrow \mathbb{R}$. These two Haar-like wavelets clearly satisfy the conditions of Eq.(1). Assuming that when $x(t)$ ($0 \leq t \leq d-1$) is a signal with d elements, the wavelet ψ is used and wavelet transform is performed at each scale $a = 1$. We define the shift parameter as $b_j = j - \delta(j)$ ($j = 0, 2, \dots, d-2$). Here, $\delta(j)$ is a monotonically increasing (or monotonically decreasing) function such that $0 \leq j \leq d-1$ and $0 \leq \delta(j) < 1$. Also, i can be expressed in terms of j as $i = \lceil \frac{j+1}{2} \rceil$.

$$\begin{bmatrix} W(1, b_0) \\ W(1, b_2) \\ \vdots \\ W(1, b_{d-2}) \end{bmatrix} = \begin{bmatrix} \cos f_1 & -\sin f_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \cos f_2 & -\sin f_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \cos f_{\frac{d}{2}} & -\sin f_{\frac{d}{2}} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(d-2) \\ x(d-1) \end{bmatrix}. \quad (8)$$

Here, we define $f_i = f(\delta(j))$. If we let x be the query q^m and define f such that $f(\delta(j)) = m\theta_i$, where $\theta_i = 10000^{-2(i-1)/d}$ and $i \in [1, 2, \dots, d/2]$, then Eq. (8) becomes exactly the same as Eq. (6) of RoPE. In other words, RoPE can be seen as a wavelet transform using Haar-like wavelets that change amplitude on a fixed scale. Furthermore, the same result as RoPE in odds dimensions can be obtained when using ψ' for wavelet transformation.¹ This wavelet transform in RoPE is performed across the number of query head dimensions d . Therefore, RoPE can be considered a wavelet transformation along the head dimension using a wavelet with a fixed scale of 2.²

¹Additionally, when $\sin m\theta_i = \cos m\theta_i$, the Haar wavelet matrix and RoPE are the same when the scale is 2, and the shift is $[2, 4, \dots, d/2]$. Refer to Appendix A.2 for the detailed proof.

²From previous research (Tancik et al., 2020), we also hypothesized that this could be equivalent to a Fourier transform. However, this hypothesis does not hold. Refer to Appendix A.3 for details.

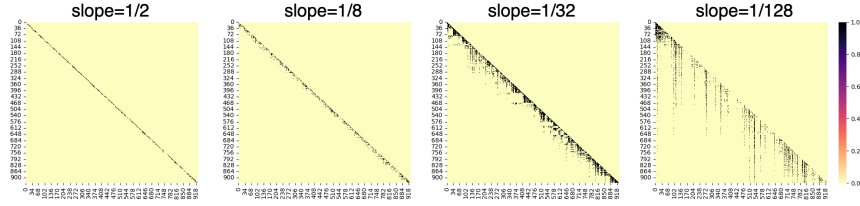


Figure 2: Heatmap of scaled attention scores via softmax normalization in ALiBi without non-overlapping inference. The vertical axis represents the query, while the horizontal axis corresponds to the key in the attention map. For clarity, values of 0.001 or more are mapped to black, while values below that are mapped to yellow. The maximum allowable length of sequences is $L_{\text{train}} = 512$ and the inference length is 1012.

4 WINDOW SIZE VARIABILITY IN ALiBi

ALiBi has a restricted receptive field and behaves in the manner of windowed attention (Chi et al., 2023; Beltagy et al., 2020). A receptive field refers to the specific region of the input space that significantly influences the model’s output, typically representing the area where the most relevant features are captured. ALiBi is expressed as follows:

$$\text{softmax}(q_m K^T + \text{slope} \cdot [-(m-1), \dots, -2, -1, 0]), \quad (9)$$

where the *slope* is a head-specific slope fixed before training and $K^T \in \mathbb{R}^{m \times d}$ is the first m keys. In this section, we analyzed the window size in ALiBi using the attention map.

4.1 INSIGHTS FROM ATTENTION MAP ANALYSIS

A heatmap of scaled attention scores obtained through softmax normalization is shown in Figure 2. The number of heads N is 8, and the slope of ALiBi is $[\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}, \frac{1}{256}]$. In extrapolation, sequences are often divided, but in this section the sequences are not divided. Experimental setting was set to the same as Section 6.1. The perplexity results are shown in Table 2.

The attention map shows that ALiBi uses multiple window sizes corresponding to relative positions and that the window size increases as the slope decreases. Moreover, previous research (Chi et al., 2023) shows that constraining the window size (the slope) to a single value leads to increased perplexity. Thus, one of the reasons ALiBi is effective, compared to a previous relative position using fixed window sizes in T5 Bias (Raffel et al., 2020), is its ability to accommodate multiple window sizes. ALiBi does not perform calculations like those in Eq. (3), so it does not exactly match the wavelet transform. However, having windows of various sizes is similar to the role of the scale parameter used in wavelet transforms.

5 WAVELET-BASED POSITIONAL REPRESENTATION

Wavelet transform (WT) is a method of analyzing signals using variable-scale wavelets, and it is possible to adjust the scale of the window. This scalability allows both broad and fine signal features to be efficiently extracted by shifting the wavelet while changing the window size. In particular, this is suitable for investigating non-stationary signals. For this reason, we believe that the wavelet transform approach is effective for capturing the dynamic fluctuations of signals that change over time, and it is also effective for the fluid nature of natural language, which is not constrained by periodicity. Furthermore, when extrapolating, it is important to be able to respond flexibly to changes in context and information. For this reason, we believe that the wavelet transform is also an effective method for extrapolation.

When applying wavelet transforms to positional encoding, a key question arises: which features should be leveraged for handling long-context dependencies? Table 1 provides an unified view of conventional positional encoding methods. Notably, RoPE shares conceptual similarities with the wavelet transform (Section 3); however, RoPE depends on absolute positional information, which

Table 1: Unified view of positional encoding method

| PE | pos | Wavelet Transform | Window Size | Receptive Field | Extrapolation |
|----------------|-----|-----------------------------------|-------------------------------|-----------------|---------------|
| RoPE | abs | Yes, $a = 2$ | L_{train} , constant | unlimited 😊 | No 😞 |
| ALiBi | rel | No | multiple | limited 😞 | Yes 😊 |
| RPE | rel | No | 32 or 128 constant | - | No 😞 |
| Wavelet (Ours) | rel | Yes, $a = [2^0, 2^1, \dots, 2^s]$ | multiple | unlimited 😊 | Yes 😊 |

limits its effective context window to the training length (L_{train}) and restricts its extrapolation capabilities. In contrast, ALiBi offers extrapolation capabilities by utilizing relative position and supports varying window sizes (Section 4). However, ALiBi’s linear bias constrains its receptive field, making it insufficient for capturing long-range dependencies. According to Press et al. (2022), conventional relative positional encoding (RPE) methods (Shaw et al., 2018; Raffel et al., 2020), which rely on a fixed window size, are similarly ineffective for extrapolation. In conclusion, we adopt relative position with flexible window sizes to handle long-context and extrapolation.

Therefore, we propose positional representation based on wavelet transform with the following characteristics:

1. **Position-based Transformation:** RoPE predominantly relies on independent transformation based on the ‘head’ dimensions. ALiBi employs multiple windows based on the relative position of the sentence, rather than the dimension of the head, which may contribute to its performance. Therefore, we apply a wavelet transform based on the relative position of the sentence.
2. **Type of Wavelets:** RoPE can be thought of as a wavelet transform using the Haar wavelet, which is the simplest wavelet. However, Haar wavelets might fall short in capturing the intricacies of natural languages. Transitioning toward the use of more sophisticated wavelet functions could enhance our approach to distill and represent a broader spectrum of features inherent in natural languages.
3. **Diversification of Window Sizes (Scale Parameters):** From the analysis of ALiBi, it was found that having multiple windows is effective for long contexts. The original of RoPE works with a single fixed scale. To address this limitation, we introduce a variety of scale and shift parameters.

5.1 METHODOLOGY

Incorporating Wavelet Transform into PE Due to the wavelet shift feature, we adopt relative position representation as ALiBi because it is more suitable than absolute position representation. In a transformer model (Vaswani et al., 2017), the self-attention mechanism operates by projecting the input sequence into three distinct representations—queries (Q), keys (K), and values (V)—using learnable weight matrices. Self-attention sublayers employ N attention heads. In self-attention sublayers, $e_{m,n}$ is the attention score for each query, and then the key is calculated. RPE (Shaw et al., 2018) expresses position by calculating the inner product of the query and the relative position embedding. We incorporate the wavelet function into RPE as follows.

$$e_{m,n} = \frac{q_m k_n^T + q_m (p_{m,n})^T}{\sqrt{d}}, \quad (10)$$

where q_m is the m th query ($q_m \in \mathbb{R}^{1 \times d}, 1 \leq m \leq L$) of a sentence of length L and k_n is the n th key ($k_n \in \mathbb{R}^{1 \times d}, 1 \leq n \leq L$) for q_m . d is the number of dimensions of each head. Here, $p_{m,n}$ is the relative position from the m -th query to the n -th key. RPE (Shaw et al., 2018) use learnable embedding for $p_{m,n} \in \mathbb{R}^d$ and fixed scale by clipping. However, instead of using learnable embeddings to represent $p_{m,n}$, we use d -pattern wavelet functions with multiple scales to calculate the position. In our method, there is no clipping, and the distance of the position expression is fixed regardless of the length of the sentence.

Wavelet Function In conventional wavelets, such as in Eq. (2), the amplitude also varies depending on the scale parameter a . In the proposed method, all amplitudes are the same.

$$\psi_{a,b}(t) = \psi\left(\frac{t-b}{a}\right). \quad (11)$$

The variable t is assigned the relative position, which is $t = m - n$. We used the Ricker Wavelet (Ricker, 1944) as a base wavelet, and it is formulated as follows.

$$\psi(t) = (1 - t^2) \exp\left(\frac{-t^2}{2}\right). \quad (12)$$

Shift and scale parameters We use s distinct patterns for the scale parameter a and $\frac{d}{s}$ patterns for the shift parameter b .

$$(a, b) \in \{2^0, 2^1, 2^2, \dots, 2^{s-1}\} \times \{0, 1, 2, 3, \dots, \frac{d}{s} - 1\}. \quad (13)$$

The scale parameter is a power of 2 derived from the principles of the discrete wavelet transform. By combining the $\frac{d}{s}$ -pattern shift parameters b with the s -pattern scale parameters a , we generate d distinct wavelets. In this way, our method can set the s -pattern **context window size** using the scale parameter a , and d -pattern **context window** using both the scale parameter a and the shift parameter b . For instance, with a head dimension of $d = 128$, we use $s = 8$ scale variants ($a \in \{2^0, 2^1, \dots, 2^7\}$) and 16 shift variants ($b \in \{0, 1, 2, \dots, 15\}$), resulting in $8 \times 16 = 128$ unique wavelets. Finally, $p_{m,n}$ is computed as follows.³

$$p_{m,n} = \left(1 - \left(\frac{m-n-b}{a}\right)^2\right) \exp\left(-\frac{1}{2}\left(\frac{m-n-b}{a}\right)^2\right). \quad (14)$$

6 SHORT-CONTEXT EXPERIMENT

6.1 EXPERIMENTAL SETTINGS

First, we conducted a small-scale experiment to compare it with various position encodings. We used the WikiText-103 dataset (Merity et al., 2017), which consists of over 103 million tokens of English Wikipedia articles. We performed a comparative evaluation using a Transformer-based language model (Baevski & Auli, 2019). The dimensionality of the word embedding d_{model} is 1024, the number of heads N is 8, the dimensionality of the heads d is 128, and the number of layers is 16. The implementation was based on the fairseq (Ott et al., 2019)-based code⁴ provided in a previous work (Press et al., 2022), and all hyperparameters were set to the same values as those in the literature (Press et al., 2022).⁵ The maximum allowable length of sequences was set to $L_{train} = 512$ and $L_{train} = 1024$.

Compared Methods Although $\theta = 10,000$ is usually used for RoPE, it has been found that extending θ to 500,000 is effective for long contexts (Xiong et al., 2024). Therefore, we compare both $\theta = 10,000$ and $\theta = 500,000$. In addition to ALiBi and RoPE, the following position representations are also compared: NoPE (Kazemnejad et al., 2023), which position information is given, and TransXL (Dai et al., 2019), which is relative positional representation uses sine waves.

Evaluation Metric We use perplexity as an evaluation metric. Following previous research (Press et al., 2022), we evaluate the validation set. To evaluate sequences longer than L_{train} tokens, it is common to divide the sequence into L_{train} -length sub-sequences, evaluate each independently, and report the average score. However, methods that use relative positions to express a wide range, such as ALiBi, Trans-XL, and the proposed method, have the potential to consider a wider range of contexts than L_{train} . For this reason, in this paper, we report not only the perplexity of non-overlapping inference but also the normal perplexity when the sequence is not divided into partial

³Implementation tips for reducing the memory and computational efficiency of the proposed method are included in Appendix A.4.

⁴https://github.com/ofirpress/attention_with_linear_biases

⁵See Appendix A.5 for more details of hyperparameters

Table 2: Perplexity of validation set in extrapolation experiments using Wikitext-103. Maximum allowable length of sequences in pre-training is $L_{\text{train}} = 512$ and $L_{\text{train}} = 1024$.

| | | Sequence Length | | | | | | | | | |
|---|------|--------------------------|--------------|--------------|--------------|--------------|--------------|---------------------------|--------------|--------------|--------------|
| | | $L_{\text{train}} = 512$ | | | | | | $L_{\text{train}} = 1024$ | | | |
| | pos | 128 | 256 | 512 | 1012 | 1512 | 2512 | 1024 | 1524 | 3024 | 5024 |
| Perplexity in Non-overlapping Inference with L_{train} | | | | | | | | | | | |
| NoPE | none | 26.38 | 23.23 | 21.53 | 21.52 | 21.53 | 21.53 | 20.81 | 21.52 | 21.49 | 21.45 |
| RoPE $\theta = 10k$ | abs | 23.82 | 20.98 | 19.39 | 19.35 | 19.39 | 19.38 | 18.42 | 19.51 | 19.52 | 19.48 |
| RoPE $\theta = 0.5m$ | abs | 23.81 | 20.95 | 19.35 | 19.32 | 19.35 | 19.33 | 18.50 | 19.53 | 19.54 | 19.50 |
| Trans-XL | rel | 24.16 | 21.53 | 19.96 | 19.92 | 19.93 | 19.96 | 18.67 | 19.75 | 19.74 | 19.70 |
| ALiBi | rel | 24.18 | 21.32 | 19.69 | 19.64 | 19.69 | 19.64 | 18.66 | 19.64 | 19.65 | 19.62 |
| Wavelet(Ricker) $a = 2^7$ | rel | 23.64 | 20.82 | 19.19 | 19.15 | 19.17 | 19.20 | 18.26 | 19.30 | 19.34 | 19.26 |
| Perplexity without Non-overlapping Inference | | | | | | | | | | | |
| NoPE | none | 26.38 | 23.23 | 21.53 | 21.03 | 21.58 | 48.48 | 20.81 | 20.45 | 22.11 | 59.37 |
| RoPE $\theta = 10k$ | abs | 23.82 | 20.98 | 19.39 | 23.25 | 44.38 | 93.94 | 18.42 | 18.29 | 33.20 | 122.52 |
| RoPE $\theta = 0.5m$ | abs | 23.81 | 20.95 | 19.35 | 23.70 | 40.39 | 77.90 | 18.50 | 18.30 | 29.25 | 83.43 |
| Trans-XL | rel | 24.16 | 21.53 | 19.96 | 19.09 | 18.92 | 19.05 | 18.67 | 18.25 | 18.17 | 18.76 |
| ALiBi | rel | 24.18 | 21.32 | 19.69 | 18.71 | 18.42 | 18.41 | 18.66 | 18.14 | 17.86 | 17.88 |
| Wavelet(Ricker) $a = 2^7$ | rel | 23.64 | 20.82 | 19.19 | 18.23 | 18.00 | 17.99 | 18.26 | 17.13 | 17.14 | 17.44 |
| Haar $a = 2^0$ | rel | 24.98 | 22.07 | 20.49 | 51.61 | 116.87 | 299.26 | - | - | - | - |
| Haar $a = 2^7$ | rel | 23.73 | 20.89 | 19.27 | 18.34 | 18.11 | 18.17 | - | - | - | - |
| Morlet $a = 2^7$ | rel | 24.15 | 21.28 | 19.65 | 19.02 | 20.46 | 26.56 | - | - | - | - |
| Gaussian $a = 2^7$ | rel | 23.77 | 20.90 | 19.30 | 18.31 | 18.02 | 17.88 | - | - | - | - |

sequences. Note that when the sequence length is less than L_{train} , the scores for the perplexity of non-overlapping inference and the normal perplexity without division into partial sequences are the same. Of course, when perplexity is considered without division into partial sequences, the performance of RoPE is expected to decrease greatly because unknown values are used for RoPE when processing a sequence longer than the length encountered during training.

6.2 MAIN RESULTS

The experimental results are shown in Table 2. The results of perplexity in inference without overlap show that the proposed method using wavelets achieved the lowest perplexity and was also effective for extrapolation. In RoPE, the values used during training are also used in inference without overlap, so the perplexity remains low even when the sequence length exceeds L_{train} . At the same time, however, perplexity is higher for ALiBi and Trans-XL than for RoPE, which is attributed to the limited context range of the position representation’s applicability due to the division of the sequence into sub-sequences. In contrast, the proposed method maintains low perplexity even in the case of division into sub-sequences, suggesting that the wavelet position representation is highly effective.

On the other hand, perplexity without non-overlapping inference showed the opposite results. First, since RoPE uses absolute positions, it is necessary to use new values for unknown positions, and thus perplexity increased significantly. However, in the case of $\theta = 500,000$, the increase in perplexity was relatively small. On the contrary, Trans-XL and ALiBi, which use relative positions, were able to handle longer contexts, and perplexity decreased as the range of position representations expanded. In the proposed method, perplexity also decreased and the best score was achieved. Trans-XL uses a position representation based on a periodic sine wave function, but the proposed method, which uses wavelets, could further decrease perplexity. This result supports our claim (see section 5) that an approach like wavelet transformation is more effective than periodic functions in capturing the fluid nature of natural language, which is not constrained by periodicity.

6.3 ANALYSIS

6.3.1 HOW EFFECTIVE ARE THE OTHER WAVELET TYPES?

We also conducted experiments to see whether the same effect could be obtained with other wavelets. The wavelets tested were the Gaussian-based wavelet $\psi(t) = \exp(-t^2)$, the Morlet-based wavelet $\psi(t) = \exp(-t^2)\cos(at)$, and the Haar-based wavelet. Note that when $\psi(t/a)$ exists in our Morlet wavelet, the frequency of this cosine wavelet is not affected by the scale parameter a .

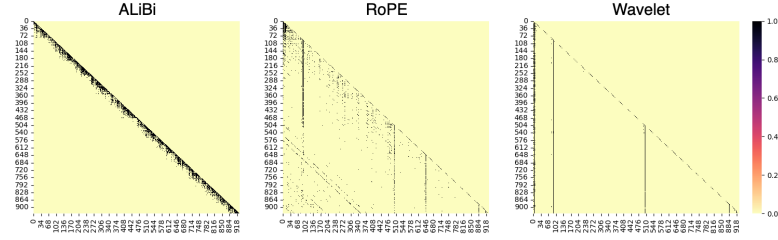


Figure 3: Heatmap of scaled attention scores via softmax normalization in 4th head after softmax operation without non-overlapping inference. The vertical axis represents the query, while the horizontal axis corresponds to the key. For clarity, values of 0.001 or more are mapped to black, while values below that are mapped to yellow. The maximum allowable length of sequences in pre-training is $L_{\text{train}} = 512$ and the inference length is 1012. See Appendix A.9 for other heads.

We used the following formula for the Haar wavelet.

$$\psi(t) = \begin{cases} 1 & -0.5 \leq t < 0, \\ -1 & -1 \leq t < -0.5, \\ 0 & \text{otherwise.} \end{cases} \quad (15)$$

We kept the shift and scale parameters constant, only changing the wavelet function. We also tested the Haar wavelet when set to $a \in \{2^0, 2^0, 2^0, \dots, 2^0\}$. Consequently, this restricted Haar wavelet had the same scale parameter setting as the RoPE demonstrated in Section 3.2.⁶ The graphs of these wavelet functions are shown in Appendix A.8 (Fig. 6). Extrapolation experiments were conducted under the same conditions as the experimental setup in Section 6, with $L_{\text{train}} = 512$ during training.

The results are shown in Table 2. The Ricker-, Haar- and Gaussian-based wavelets had lower perplexity than the Morlet wavelet. It is possible that complex wavelets with multiplied cosine waves, such as Morlet wavelets, are not suitable for relative positional representation. On the other hand, wavelets with all positive values, such as Gaussian-based wavelets, are expected to represent positions within a narrower distance than the window specified by the scale parameter due to softmax normalization. This suggests that wavelets with a specific range of negative values are suitable, like a Ricker wavelet, for positional representation. Although the Haar wavelet is simple, it is such a wavelet with negative values within a specific range. Therefore, it is considered effective, although not as much as a Ricker wavelet. However, when the scale parameter is restricted ($a \in \{2^0, \dots, 2^0\}$), as in RoPE, the perplexity increases. This demonstrates the importance of having multiple scales, or in this case, window sizes.

6.3.2 CAN IT HANDLE TOKENS WITH LONG-RANGE DEPENDENCIES?

Figure 3 shows the attention map of scaled attention scores obtained through softmax normalization for the proposed method. The inference length is $L = 1012$ without non-overlapping inference. The primary notable feature of the proposed method is that it is always able to attend to specific tokens. The words that always receive attention are those that are important in the sentence, such as the special token, the first token, and the subject of the sequence. On the other hand, ALiBi has a restricted receptive field for attention, making it unable to capture long-distance dependencies. Similar to the proposed method, RoPE emphasizes important and special words but it struggles to capture those that are farther apart. Moreover, as the sentence lengthens, it loses the ability to attend to the initial word. This tendency was also seen in sentences shorter than L_{train} . Accordingly, the proposed method has demonstrated its superiority at capturing long dependencies without restricting the receptive field of attention.

⁶Normally, the wave is localized when $t > 0$ in the Haar wavelet, but in the decoder model, only the range $t < 0$ is used. Therefore, we transformed the Haar wavelet into a form that reflects the original function $f(x)$ across the y-axis.

Table 3: Perplexity in Non-overlapping Inference with $L_{\text{train}} = 4096$.

| | Sequence Length | | | | |
|-----------------------|-----------------|-------------|-------------|-------------|-------------|
| | 2 k | 4 k | 8 k | 16 k | 32 k |
| RoPE $\theta = 0.5$ m | 8.37 | 7.85 | 9.33 | 9.12 | 8.90 |
| Wavelet $a = 2^9$ | 8.21 | 7.35 | 9.01 | 8.83 | 8.60 |

7 LONG CONTEXT

7.1 EXPERIMENTAL SETTINGS

Next, we conducted a large-scale experiment using a Llama-based model (Touvron et al., 2023b). We pre-trained the Llama-2-7B⁷ model from scratch. For pre-training, we used the RedPajama dataset (Computer, 2023), which selects a 1B-token sample of all samples.⁸ The maximum allowable length of sequences in pre-training was set to $L_{\text{train}} = 4096$. For the same reason as given in Section 6.1, we set $\theta = 500,000$ for RoPE. Furthermore, when the scale parameter is $a \in \{2^0, 2^1, \dots, 2^7\}$, the range within which the wavelet is localized becomes narrow. Therefore, in our method, we changed the scale parameter to $a \in \{2^2, 2^3, \dots, 2^9\}$. The other parameters are the same as those for the Llama-2-7B model (Touvron et al., 2023b). We used CodeParrot⁹ for evaluation, which is good for long-distance testing because it requires an understanding of patterns and contextualization of information over long distances. In accordance with previous research (Rubin & Berant, 2024; Wu et al., 2022; Zhang et al., 2024), we then used 100 sampled sequences in the training set for evaluation. In this experiment, due to the large model size and long sequence length, we report perplexity only for non-overlapping inference using L_{train} , since the memory capacity is exceeded.

7.2 MAIN RESULTS

The experimental results are shown in Table 3. Regardless of interpolation or extrapolation, the perplexity of our method was lower than RoPE. Therefore, even with large-scale models and long contexts, our method was found to be effective. Furthermore, the results in Section 6.2 show that not dividing the sequence further reduces perplexity. Therefore, there is a possibility that our method will further reduce perplexity.

8 CONCLUSION

In this paper, we demonstrated that RoPE can be interpreted as a wavelet transform, and we introduced a novel positional representation method that leverages the wavelet transform’s advantages, effectively capturing positional information across various window sizes. Our experimental results demonstrate the proposed method’s superior performance in extrapolation tasks when compared to traditional positional representation techniques. Importantly, our approach offers the advantage of not constraining the receptive field, which allows more flexible and comprehensive analysis of positions. It is known that calculating relative positions requires more resources than calculating absolute positions, so we show methods for reducing memory consumption in Appendix A.4. Nevertheless, the overhead imposed by calculating relative positions can still be a bottleneck, so reducing the amount of required calculation remains a future work.

REFERENCES

Alexei Baevski and Michael Auli. Adaptive input representations for neural language modeling. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=ByxZX20qFQ>.

⁷<https://huggingface.co/meta-llama/Llama-2-7b>

⁸See Appendix A.6 for more details of the hyperparameters.

⁹<https://huggingface.co/datasets/codeparrot/codeparrot-clean>

- Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, 2020.
- Alexandre Bernardino and José Santos-Victor. A real-time gabor primal sketch for visual attention. In Jorge S. Marques, Nicolás Pérez de la Blanca, and Pedro Pina (eds.), *Pattern Recognition and Image Analysis*, pp. 335–342, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-32237-5.
- bloc97. Ntk-aware scaled rope allows llama models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation., 2023. URL https://www.reddit.com/r/LocalLLaMA/comments/141z7j5/ntkaware_scaled_rope_allows_llama_models_to_have/.
- Ronald Newbold Bracewell and Ronald N Bracewell. *The Fourier transform and its applications*, volume 31999. McGraw-Hill New York, 1986.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf.
- Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation, 2023. URL <https://arxiv.org/abs/2306.15595>.
- Ta-Chung Chi, Ting-Han Fan, Alexander Rudnicky, and Peter Ramadge. Dissecting transformer length extrapolation via the lens of receptive field analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13522–13537, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.756. URL <https://aclanthology.org/2023.acl-long.756>.
- Together Computer. Redpajama: An open source recipe to reproduce llama training dataset, 2023. URL <https://github.com/togethercomputer/RedPajama-Data>.
- Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2978–2988, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1285. URL <https://aclanthology.org/P19-1285>.
- Ingrid Daubechies. Ten lectures on wavelets. *Society for industrial and applied mathematics*, 1992.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL <https://aclanthology.org/N19-1423>.
- W. M. Gentleman and G. Sande. Fast fourier transforms: for fun and profit. In *Proceedings of the November 7-10, 1966, Fall Joint Computer Conference, AFIPS '66 (Fall)*, pp. 563–578, New York, NY, USA, 1966. Association for Computing Machinery. ISBN 9781450378932. doi: 10.1145/1464291.1464352. URL <https://doi.org/10.1145/1464291.1464352>.

- A. Grossmann and J. Morlet. Decomposition of hardy functions into square integrable wavelets of constant shape. *SIAM Journal on Mathematical Analysis*, 15(4):723–736, 1984. doi: 10.1137/0515056. URL <https://doi.org/10.1137/0515056>.
- A. Haar. Zur theorie der orthogonalen funktionensysteme. (erste mitteilung). *Mathematische Annalen*, 69:331–371, 1910. URL <http://eudml.org/doc/158469>.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. Mistral 7b, 2023. URL <https://arxiv.org/abs/2310.06825>.
- Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. The impact of positional encoding on length generalization in transformers. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 24892–24928. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/4e85362c02172c0c6567ce593122d31c-Paper-Conference.pdf.
- Gregory R. Lee, Ralf Gommers, Filip Waselewski, Kai Wohlfahrt, and Aaron O8217;Leary. Py-wavelets: A python package for wavelet analysis. *Journal of Open Source Software*, 4(36):1237, 2019. doi: 10.21105/joss.01237. URL <https://doi.org/10.21105/joss.01237>.
- Yang Li, Si Si, Gang Li, Cho-Jui Hsieh, and Samy Bengio. Learnable fourier features for multi-dimensional spatial positional encoding. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=R0h3NUMao_U.
- Xiaoran Liu, Hang Yan, Chenxin An, Xipeng Qiu, and Dahua Lin. Scaling laws of roPE-based extrapolation. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=JO7k0SJ5V6>.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=Bkg6RiCqY7>.
- S.G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989. doi: 10.1109/34.192463.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL <https://openreview.net/forum?id=Byj72udxe>.
- Nhat Khang Ngo, Truong Son Hy, and Risi Kondor. Multiresolution graph transformers and wavelet positional encoding for learning long-range and hierarchical structures. *The Journal of Chemical Physics*, 159(3):034109, 07 2023a. ISSN 0021-9606. doi: 10.1063/5.0152833. URL <https://doi.org/10.1063/5.0152833>.
- Nhat Khang Ngo, Truong Son Hy, and Risi Kondor. Multiresolution graph transformers and wavelet positional encoding for learning hierarchical structures. *arXiv preprint arXiv:2302.08647*, 2023b.
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.
- Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=wHBfxhZulu>.
- Ofir Press, Noah Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=R8sQPpGCv0>.

- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL <http://jmlr.org/papers/v21/20-074.html>.
- Norman Ricker. Wavelet functions and their polynomials. *Geophysics*, 9(3):314–323, 07 1944. ISSN 0016-8033. doi: 10.1190/1.1445082. URL <https://doi.org/10.1190/1.1445082>.
- Ohad Rubin and Jonathan Berant. Retrieval-pretrained transformer: Long-range language modeling with self-retrieval, 2024. URL <https://arxiv.org/abs/2306.13421>.
- Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 464–468, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2074. URL <https://aclanthology.org/N18-2074>.
- Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2021.
- Yutao Sun, Li Dong, Barun Patra, Shuming Ma, Shaohan Huang, Alon Benhaim, Vishrav Chaudhary, Xia Song, and Furu Wei. A length-extrapolatable transformer. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14590–14604, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.816. URL <https://aclanthology.org/2023.acl-long.816>.
- Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS ’20*, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.
- R. Tian, Z. Wu, Q. Dai, H. Hu, Y. Qiao, and Y. Jiang. Resformer: Scaling vits with multi-resolution training. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22721–22731, Los Alamitos, CA, USA, jun 2023. IEEE Computer Society. doi: 10.1109/CVPR52729.2023.02176. URL <https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.02176>.
- 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. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971, 2023a. URL <https://api.semanticscholar.org/CorpusID:257219404>.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023b. URL <https://arxiv.org/abs/2307.09288>.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von

Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Benyou Wang, Donghao Zhao, Christina Lioma, Qiuchi Li, Peng Zhang, and Jakob Grue Simonsen. Encoding word order in complex embeddings. In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=Hke-WTVtwr>.

Yuhuai Wu, Markus Norman Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=TrjbxzRcnf->.

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, Madian Khabsa, Han Fang, Yashar Mehdad, Sharan Narang, Kshitiz Malik, Angela Fan, Shruti Bhosale, Sergey Edunov, Mike Lewis, Sinong Wang, and Hao Ma. Effective long-context scaling of foundation models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4643–4663, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.260. URL <https://aclanthology.org/2024.naacl-long.260>.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. Soaring from 4k to 400k: Extending llm’s context with activation beacon, 2024. URL <https://arxiv.org/abs/2401.03462>.

A APPENDIX

A.1 ROTARY POSITION EMBEDDING

RoPE incorporates positional information directly into the self-attention mechanism by rotating the query and key vectors in the complex space. When divided into even and odd dimensions, the following calculations are performed for the m -th query in each sequence. In even dimensions, RoPE is expressed as follows.

$$\begin{bmatrix} q_0^m \\ q_2^m \\ \vdots \\ q_{d-2}^m \end{bmatrix} = \begin{bmatrix} \cos m\theta_1 & -\sin m\theta_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \cos m\theta_2 & -\sin m\theta_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \cos m\theta_{d/2} & -\sin m\theta_{d/2} \end{bmatrix} \begin{bmatrix} q_0^m \\ q_1^m \\ q_2^m \\ q_3^m \\ \vdots \\ q_{d-2}^m \\ q_{d-1}^m \end{bmatrix}. \quad (16)$$

In odds dimensions, RoPE is expressed as follows.

$$\begin{bmatrix} q_1^m \\ q_3^m \\ \vdots \\ q_{d-1}^m \end{bmatrix} = \begin{bmatrix} \sin\theta_1 & \cos\theta_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \sin m\theta_2 & \cos m\theta_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \sin m\theta_{d/2} & \cos m\theta_{d/2} \end{bmatrix} \begin{bmatrix} q_0^m \\ q_1^m \\ q_2^m \\ q_3^m \\ \vdots \\ q_{d-2}^m \\ q_{d-1}^m \end{bmatrix}. \quad (17)$$

where $q^m \in \mathbb{R}^{1 \times d}$ is the m -th query when the number of dimensions is d and $\theta_i = 10000^{-2(i-1)/d}$, $i \in [1, 2, \dots, d/2]$. The same process is also performed for the n -th key $k^n \in \mathbb{R}^{1 \times d}$.

A.2 HAAR WAVELET

Here, we explain wavelet transform using the Haar wavelet, which is the simplest wavelet. The definition of the Haar wavelet is as follows.

$$\psi(t) = \begin{cases} 1 & 0 \leq t < 1/2, \\ -1 & 1/2 \leq t < 1, \\ 0 & \text{otherwise.} \end{cases} \quad \phi(t) = \begin{cases} 1 & 0 \leq t < 1, \\ 0 & \text{otherwise.} \end{cases} \quad (18)$$

Haar wavelets are defined not only by a wavelet function ψ but also by a scaling function ϕ .

The method of analyzing signals by performing a discrete wavelet transform using these two functions is called multi-resolution analysis. When the scale is fixed at 2 and the shift $b \in [0, 2, \dots, d/2]$, the wavelet transform using the wavelet function and scaling function is expressed as follows.

$$\begin{bmatrix} \psi_{2,0}(0) & \psi_{2,0}(1) & \psi_{2,0}(2) & \psi_{2,0}(3) & \dots & \psi_{2,0}(T-2) & \psi_{2,0}(T-1) \\ \phi_{2,0}(0) & \phi_{2,0}(1) & \phi_{2,0}(2) & \phi_{2,0}(3) & \dots & \phi_{2,0}(T-2) & \phi_{2,0}(T-1) \\ \psi_{2,0}(-2) & \psi_{2,0}(-1) & \psi_{2,0}(0) & \psi_{2,0}(1) & \dots & \psi_{2,0}(T-4) & \psi_{2,0}(T-3) \\ \phi_{2,0}(-2) & \phi_{2,0}(-1) & \phi_{2,0}(0) & \phi_{2,0}(1) & \dots & \phi_{2,0}(T-4) & \phi_{2,0}(T-3) \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ \psi_{2,0}(-\frac{d}{2}) & \psi_{2,0}(-\frac{d}{2}+1) & \psi_{2,0}(-\frac{d}{2}+2) & \psi_{2,0}(-\frac{d}{2}+3) & \dots & \psi_{2,0}(0) & \psi_{2,0}(1) \\ \phi_{2,0}(-\frac{d}{2}) & \phi_{2,0}(-\frac{d}{2}+1) & \phi_{2,0}(-\frac{d}{2}+2) & \phi_{2,0}(-\frac{d}{2}+3) & \dots & \phi_{2,0}(0) & \phi_{2,0}(1) \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(T-2) \\ x(T-1) \end{bmatrix}. \quad (19)$$

From Eq.(18), $\psi_{2,0}$ and $\phi_{2,0}$ are as follows.

$$\psi_{2,0}(t) = \begin{cases} 1/\sqrt{2} & 0 \leq t < 1, \\ -1/\sqrt{2} & 1 \leq t < 2, \\ 0 & \text{otherwise.} \end{cases} \quad \phi_{2,0}(t) = \begin{cases} 1/\sqrt{2} & 0 \leq t < 2, \\ 0 & \text{otherwise.} \end{cases} \quad (20)$$

Therefore, the Haar wavelet transform is a 2×2 block matrix.

$$\begin{bmatrix} \psi(2,0) \\ \phi(2,0) \\ \psi(2,2) \\ \phi(2,2) \\ \vdots \\ \psi(2,T-2) \\ \phi(2,T-2) \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 & \dots & 0 & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} & \dots & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & 1/\sqrt{2} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1/\sqrt{2} & -1/\sqrt{2} \\ 0 & 0 & 0 & 0 & \dots & 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ \vdots \\ x(T-2) \\ x(T-1) \end{bmatrix}. \quad (21)$$

This matrix is the Haar forward transform using matrix multiplication for a T element signal. This matches the RoPE matrix with $m\theta = \pi/4$.

A.3 ISN'T ROPE A FOURIER TRANSFORM?

We also hypothesized that this could be equivalent to a Fourier transform. However, this hypothesis does not hold. When a signal $x(t)$ that changes over time is Fourier transformed, its spectrum $F(k)$ is obtained. The process of converting an actual discrete signal $x(t)$ into a spectrum $F(k)$ is as follows.

$$F(f) = \sum_{t=0}^T x(t)w^{f \cdot t} \quad (22)$$

The Fourier transform can be expressed as a matrix formula as follows.

$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ \vdots \\ F(f) \end{bmatrix} = \begin{bmatrix} w^{0 \cdot 0} & w^{0 \cdot 1} & w^{0 \cdot 2} & \dots & w^{0 \cdot (T-1)} \\ w^{1 \cdot 0} & w^{1 \cdot 1} & w^{1 \cdot 2} & \dots & w^{1 \cdot (T-1)} \\ w^{2 \cdot 0} & w^{2 \cdot 1} & w^{2 \cdot 2} & \dots & w^{2 \cdot (T-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w^{f \cdot 0} & w^{f \cdot 1} & w^{f \cdot 2} & \dots & w^{f \cdot (T-1)} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(T-1) \end{bmatrix}. \quad (23)$$

Here, $f \in \mathbb{R}$ is the wave number, $T \in \mathbb{R}$ is the number of samples, and i is the imaginary unit. $w = \exp(-\frac{2\pi i}{T})$ is called the Twiddle Factor (Gentleman & Sande, 1966), which is a complex number expressed in polar form using Euler's formula $e^{-i\theta} = \cos\theta - i\sin\theta$. In the complex plane, $w^{f \cdot t}$ represents a point on the unit circle with an argument of the complex number $-\frac{ft2\pi}{T}$. From this formula, we can see that the Fourier transform calculates the inner product of all signals and sine waves. However, in RoPE, the inner product with sine waves is calculated only within each block.

Next, when calculating the attention score with RoPE, does the Fourier transform hold? Attention scores of the m -th query q^m and the n -th key k^n with RoPE are calculated as follows.

$$\left[R_m^1 (Q_m^1)^T, \dots, R_m^{d/2} (Q_m^{d/2})^T \right] \begin{bmatrix} R_n^1 K_n^1 \\ \vdots \\ R_n^{d/2} K_n^{d/2} \end{bmatrix} = \sum_{i=1}^{d/2} (Q_m^i)^T R_{n-m}^i K_n^i, \quad (24)$$

where $Q_m^{d/2}$ is the query divided into every two dimensions, and $R_m^{d/2}$ is the rotation matrix.

$$Q_m^{d/2} = \begin{bmatrix} q_m^{d-1} \\ q_m^d \end{bmatrix}, K_n^{d/2} = \begin{bmatrix} k_n^{d-1} \\ k_n^d \end{bmatrix}, R_m^{d/2} = \begin{bmatrix} \cos m\theta_{d/2} & -\sin m\theta_{d/2} \\ \sin m\theta_{d/2} & \cos m\theta_{d/2} \end{bmatrix}.$$

To align with the Fourier transform, as illustrated in Equation 23, a process involving the inner product between a frequency tensor of dimensions $f \times T$ and a signal tensor of dimensions $T \times 1$ (such as the query vector) is required. However, RoPE operates on independent 2×2 blocks, where each block is processed separately. Consequently, RoPE's block-wise operations do not conform to the structure required by the Fourier transform. Moreover, if we focus solely on the RoPE and key operations in Equation 24, they may appear to align with the structure of a Fourier transform. However, since the final step involves taking the inner product with the query, the overall operation deviates from a perfect match with the Fourier transform. Furthermore, the rotation factor represents a rotation in the complex plane, and even if it is expressed as in Eq.(23) using a rotation matrix, it does not completely match a rotation matrix that represents a rotation in the Euclidean plane.

Therefore, RoPE cannot be equated with the Fourier transform. Furthermore, even if it were the same as the Fourier transform, it would be unsuitable for processing non-stationary signals and thus unsuitable for processing natural language, which is a non-stationary flow.

A.4 IMPLEMENTATION TIPS FOR WAVELET POSITION REPRESENTATION

Tip 1 Similar to RPE(Shaw et al., 2018), we use Eq. (10) as follows:

$$\alpha_{ij} = \text{softmax}\left(\frac{q_i K^T + q_i (p_{ij})^T}{\sqrt{d_k}}\right).$$

By transforming it in this way, it is possible to reduce the computational complexity to $O(batch \times n \times length^2 \times d + length^2 \times d)$, where $batch$ is the batch size, n is the number of heads, $length$ is the number of tokens, and d is the number of dimensions of each head. **The experiments in Section 6 are implemented based on the methodology introduced in this section.**

Tip 2 When dealing with long contexts of over 4 k with a large model, the memory efficiency of $(d, length, length)$ of the wavelet position becomes a bottleneck. Therefore, we further reduce the memory usage to $(d, length)$ by using `torch.scatter` to scatter the wavelet position representation to the attention mask. In the relative position representation in the decoder, only the position information of the token before the current token is required, for example, $0, -1, -2, \text{etc.}$. Therefore, we pre-compute the information up to $0, -1, -2, \dots, -length$ and reduce the memory usage by using `torch.scatter` to distribute it. Specifically, we prepare a $(d, length)$ wavelet tensor and calculate the 2D inner product with the query, which has been transposed to $(length \times batch, d)$. The tensor after the calculation becomes $(length \times batch, length)$, which is then scattered using `torch.scatter` so that it becomes a relative position in the attention mask. This reduces the amount of memory used from $(d, length, length)$ to $(d, length)$, and the calculation can be performed using calculations between 2D tensors. **The experiments in Section 7 are implemented based on the methodology introduced in this section.**

A.5 EXPERIMENTAL SETTINGS IN SHORT-CONTEXT EXPERIMENT

The parameter settings used in the extrapolation experiments were the same as those in the original ALiBi paper. The dimensionality of the word embedding d_{model} is 1024, the number of heads N is 8, the dimensionality of the heads d is 128, and the number of layers is 16. The implementation was based on the fairseq (Ott et al., 2019)-based code¹⁰ provided in a previous work (Press et al., 2022), and all hyperparameters were set to the same values as those in the literature (Press et al., 2022). The number of training epochs is 205, and the batch size is 9216. The learning rate was set to 1.0, and the learning process was updated by $1e-7$ every 16,000 steps.

A.6 EXPERIMENTAL SETTINGS IN LONG-CONTEXT EXPERIMENT

Pre-training The dimensionality of the word embedding d_{model} is 4096, the number of heads N is 32, the dimensionality of the heads d is 128, and the number of layers is 32. The number of training steps is 30000, and the batch size is 1. The learning rate was set to 0.0003. We used AdamW (Loshchilov & Hutter, 2019) as the optimizer, with $(\beta_1, \beta_2) = (0.9, 0.95)$.

Fine-tuning for Position Interpolation We used the RedPajama dataset (Computer, 2023) for fine-tuning. The number of training steps is 1000, and the batch size is 1. The learning rate was set to $1e-7$. The max context length is 8000. We used AdamW (Loshchilov & Hutter, 2019) as the optimizer, with $(\beta_1, \beta_2) = (0.9, 0.9)$.

¹⁰https://github.com/ofirpress/attention_with_linear_biases

A.7 RICKER WAVELET

Figure 4 shows the ricker wavelets with multiple scale a .

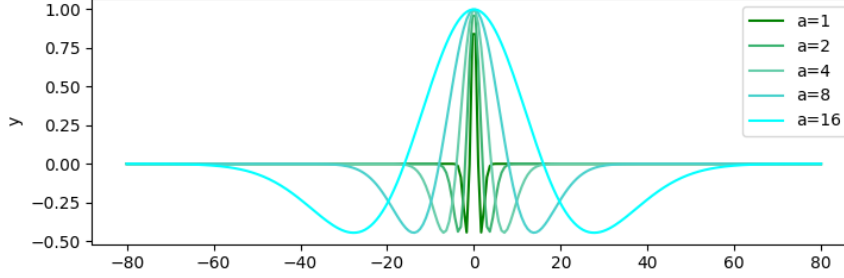


Figure 4: Graph of compared ricker wavelet functions with $a = [2^0, 2^1, 2^2, 2^3, 2^4]$

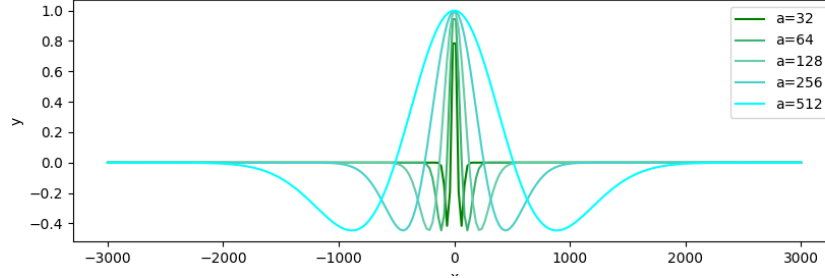


Figure 5: Graph of compared ricker wavelet functions with $a = [2^5, 2^6, 2^7, 2^8, 2^9]$

A.8 WAVELET TYPE

Figure 6 shows graphs of the wavelets compared in Section 6.3.1. It can be seen that the simplest is the Haar wavelet, while the most complex is the Morlet wavelet.

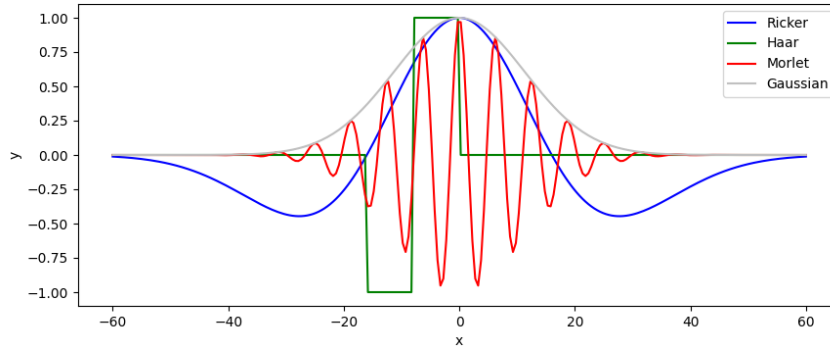


Figure 6: Graph of compared wavelet functions. The case with scale parameter $a = 2^4$ and shift parameter $b = 0$ is shown.

A.9 CAN IT HANDLE TOKENS WITH LONG-RANGE DEPENDENCIES?

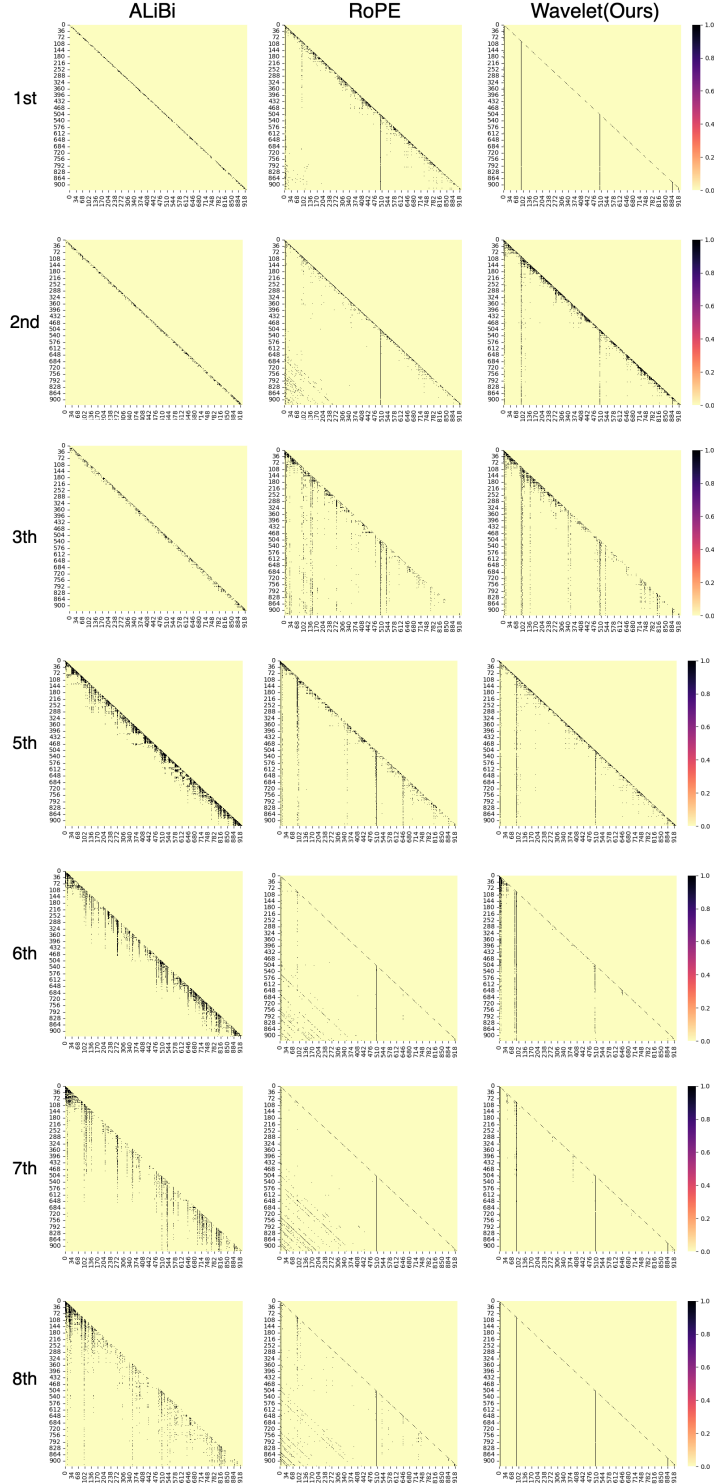


Figure 7: Heatmap of scaled attention scores via softmax normalization in 1-3 and 5-8th head after softmax operation for ALiBi, RoPE, and our method. For clarity, values of 0.001 or more are mapped to black, while values below that are mapped to yellow.

Table 4: Perplexity of validation set in extrapolation experiments using Wikitext-103. Maximum allowable length of sequences in pre-training is $L_{\text{train}} = 512$.

| | | | Sequence Length | | | | | |
|--|--------------------------------|---------------------------|-----------------|-------|-------|-------|--------|--------|
| scale a | | shift b | 128 | 256 | 512 | 1012 | 1512 | 2512 |
| Perplexity without Non-overlapping Inference | | | | | | | | |
| Ricker | $\{2^0, 2^1, \dots, 2^7\}$ | $\{0, 1, 2, \dots, 15\}$ | 23.64 | 20.82 | 19.19 | 18.23 | 18.00 | 17.99 |
| Ricker | $\{2^1, 2^2, \dots, 2^8\}$ | $\{0, 1, 2, \dots, 15\}$ | 23.77 | 20.89 | 19.25 | 18.23 | 17.97 | 18.02 |
| Ricker | $\{2^2, 2^3, \dots, 2^9\}$ | $\{0, 1, 2, \dots, 15\}$ | 23.92 | 21.03 | 19.40 | 18.41 | 18.14 | 18.07 |
| Ricker | $\{2^0, 2^1, 2^2, 2^3\}$ | $\{0, 1, 2, \dots, 31\}$ | 23.96 | 21.13 | 19.55 | 18.87 | 19.40 | 21.73 |
| Ricker | $\{2^0, 2^1\}$ | $\{0, 1, 2, \dots, 63\}$ | 24.49 | 21.60 | 19.95 | 20.90 | 32.01 | 70.80 |
| Ricker | $\{2^0, 2^1, \dots, 2^{15}\}$ | $\{0, 1, 2, \dots, 7\}$ | 23.74 | 20.88 | 19.24 | 18.22 | 17.96 | 17.84 |
| Ricker | $\{2^0, 2^1, \dots, 2^{31}\}$ | $\{0, 1, 2, 3\}$ | 23.75 | 20.86 | 19.26 | 18.24 | 17.96 | 17.84 |
| Ricker | $\{2^0, 2^1, \dots, 2^{63}\}$ | $\{0, 1\}$ | 23.75 | 20.88 | 19.30 | 18.31 | 18.04 | 18.02 |
| Ricker | $\{2^0, 2^1, \dots, 2^{127}\}$ | $\{0\}$ | 23.97 | 21.10 | 19.46 | 18.50 | 18.27 | 18.29 |
| Ricker | $\{2^7\}$ | $\{0, 1, 2, \dots, 127\}$ | 24.35 | 21.45 | 19.80 | 20.68 | 20.87 | 21.31 |
| Gaussian | $\{2^0, 2^1, \dots, 2^7\}$ | $\{0, 1, 2, \dots, 15\}$ | 23.77 | 20.90 | 19.30 | 18.31 | 18.02 | 17.88 |
| Gaussian | $\{2^1, 2^2, \dots, 2^8\}$ | $\{0, 1, 2, \dots, 15\}$ | 23.92 | 21.02 | 19.41 | 18.41 | 18.15 | 18.01 |
| Gaussian | $\{2^2, 2^3, \dots, 2^9\}$ | $\{0, 1, 2, \dots, 15\}$ | 23.98 | 21.09 | 19.46 | 18.43 | 18.13 | 17.93 |
| Gaussian | $\{2^0, 2^1, 2^2, 2^3\}$ | $\{0, 1, 2, \dots, 31\}$ | 23.83 | 20.96 | 19.33 | 18.43 | 18.40 | 18.94 |
| Gaussian | $\{2^0, 2^1\}$ | $\{0, 1, 2, \dots, 63\}$ | 24.28 | 21.35 | 19.70 | 18.96 | 19.63 | 23.14 |
| Gaussian | $\{2^0, 2^1, \dots, 2^{15}\}$ | $\{0, 1, 2, \dots, 7\}$ | 23.72 | 20.86 | 19.24 | 18.24 | 17.95 | 17.77 |
| Gaussian | $\{2^0, 2^1, \dots, 2^{31}\}$ | $\{0, 1, 2, 3\}$ | 23.78 | 20.92 | 19.29 | 18.30 | 18.01 | 17.85 |
| Gaussian | $\{2^0, 2^1, \dots, 2^{63}\}$ | $\{0, 1\}$ | 23.86 | 20.98 | 19.37 | 18.46 | 18.20 | 18.10 |
| Gaussian | $\{2^0, 2^1, \dots, 2^{127}\}$ | $\{0\}$ | 24.21 | 21.31 | 19.68 | 18.71 | 18.45 | 18.45 |
| Gaussian | $\{2^7\}$ | $\{0, 1, 2, \dots, 127\}$ | 24.48 | 21.62 | 20.05 | 19.53 | 22.63 | 35.23 |
| Haar | | | 24.98 | 22.07 | 20.49 | 51.61 | 116.87 | 299.26 |
| Haar | $\{2^0, 2^1, \dots, 2^7\}$ | $\{0, 1, 2, \dots, 15\}$ | 23.73 | 20.89 | 19.27 | 18.34 | 18.11 | 18.17 |
| Morlet | $\{2^0, 2^1, \dots, 2^7\}$ | $\{0, 1, 2, \dots, 15\}$ | 24.15 | 21.28 | 19.65 | 19.02 | 20.46 | 26.56 |

A.10 ABRATION STUDY OF SCALE AND SHIFT PARAMETER

In this section, we present the findings from our ablation study focusing on the shift and scale parameters of the Ricker and Gaussian wavelets. As indicated in Table 2, both wavelet types demonstrate substantial effectiveness in our method. To further evaluate their performance, we explored the contributions of the two parameters: the scale parameter a and the shift parameter b , while keeping all other settings consistent with those outlined in Section 6.

Results The results of our experiments are summarized in Table 4. Both the Ricker and Gaussian wavelets exhibit similar trends regarding the influence of the scale and shift parameters on extrapolation performance. Initially, we observed that increasing the scale parameter value a while holding the shift parameter b ($\{2^0, 2^1, \dots, 2^7\} \times \{0, 1, 2, \dots, 15\}$, $\{2^1, 2^2, \dots, 2^8\} \times \{0, 1, 2, \dots, 15\}$ and $\{2^2, 2^3, \dots, 2^9\} \times \{0, 1, 2, \dots, 15\}$) constant maintained the performance of extrapolation, albeit with some fluctuations. Conversely, when we increased the number of shift parameters while decreasing the number of scale parameters ($\{2^0, 2^1, 2^2, 2^4\} \times \{0, 1, 2, \dots, 31\}$ and $\{2^0, 2^1\} \times \{0, 1, 2, \dots, 63\}$), there was a noticeable decline in performance. This finding underscores the significance of the scale parameters in extrapolation. Moreover, we found that increasing the number of scale parameters while decreasing the number of shift parameters led to performance improvements in some instances ($\{2^0, 2^1, \dots, 2^{15}\} \times \{0, 1, 2, \dots, 7\}$ and $\{2^0, 2^1, \dots, 2^{31}\} \times \{0, 1, 2, 3\}$). However, when the shift parameters were reduced to two or entirely eliminated ($\{2^0, 2^1, \dots, 2^{63}\} \times \{0, 1\}$ and $\{2^0, 2^1, \dots, 2^{127}\}$), relying solely on the scale parameters resulted in a deterioration of extrapolation performance. And even when the scale parameter was fixed and only the shift parameter was used ($\{2^7\} \times \{0, 1, 2, \dots, 127\}$), the extrapolation performance decreased. This suggests a potential importance of the shift parameters as well.

In conclusion, our analysis highlights the critical roles of both shift and scale parameters in the effectiveness of our wavelet-based method.

A.1.1 ABRATION STUDY OF WAVELET TYPES

In this section, we also explored a variety of wavelet types beyond those previously discussed. In Section 6.3.1, our focus was primarily on wavelets that can be computed directly from mathematical formulas. However, in this section, we expanded our inquiry to include wavelets with varying numbers of vanishing moments as well as discrete wavelet transformations. Additionally, drawing from previous research (Wang et al., 2020), we considered the necessity for a distinct approach when incorporating complex numbers into positional encoding. Consequently, our study did not encompass wavelets that incorporate complex numbers.

Wavelet types The specific wavelets under consideration in our investigation are outlined as follows:

- Daubechies (db) (Daubechies, 1992) - Compactly supported orthonormal wavelets
- Symlets (sym) - Wavelet with minimum asymmetry
- Coiflets (coif) - The scaling and wavelet functions have the same number of vanishing moments
- Meyer (dmey) - Wavelet defined in the frequency domain
- Biorthogonal Spline (bior) - Two wavelets are used. One is used for decomposition, and the other for reconstruction.
- Reverse biorthogonal Spline (rbio)

In addition, the graphs of these wavelets are shown in Figures 8 and 9. Furthermore, as the number of vanishing moments increases, the wave oscillation becomes larger. Therefore, we also conducted a survey by vanishing point moment. The name of a wavelet is derived from the number of vanishing moments. For example, db6 is a Daubechies wavelet with 6 vanishing moments, and sym3 is a Symlet wavelet with 3 vanishing moments. In the case of Coiflet wavelets, coif3 is a Coiflet wavelet with 6 vanishing moments. The names of bior and rbio wavelets are derived from the number of vanishing moments possessed by the decomposition and reconstruction wavelets, respectively. For example, bior3.5 is Biorthogonal wavelet that has 3 vanishing moments for the decomposition wavelet and 5 vanishing moments for the reconstruction wavelet. Biorthogonal wavelets and Reverse-Biorthogonal wavelets calculate the approximate values of decomposition wavelets and reconstruction wavelets, but in this case, we only used the values of decomposition wavelets.

Experimental Settings We used Pywavelet (Lee et al., 2019)¹¹ to calculate the approximate values of these wavelets. In addition, in this experiment, we calculated the approximate values by specifying 8 levels of $\{1, 2, \dots, 8\}$ instead of the 8-pattern scale parameters $\{2^0, 2^1, \dots, 2^7\}$. We used the shift parameter $\{0, 1, 2, \dots, 15\}$. The other experimental settings are the same as in Section 6.

Results The experimental results are summarized in Table 5. Overall, the performance observed was suboptimal. However, it is important to note that the parameters were fixed at levels $\{1, 2, \dots, 8\}$, we believe that performance may be enhanced with adjustments to these levels. Notably, the rbio1.1 wavelet demonstrated promising extrapolation capabilities, suggesting significant potential for future improvements. In contrast, the coif and dmey wavelets exhibited limited performance, even with shorter sequences, indicating their potential unsuitability for position encoding tasks. Conversely, while the extrapolation performance (> 512) of other wavelets was generally low, their interpolation performance (≤ 512) remained consistently stable, highlighting another avenue for enhancement. Furthermore, the performance of the db, bior, and rbio wavelets showed a positive correlation with an increasing number of vanishing points. This finding underscores the importance of vanishing points as a critical factor influencing performance. In conclusion, our analysis indicates that both the shape of the wavelet and the number of vanishing points play significant roles in determining extrapolation performance. Future work should explore these relationships further to identify optimal configurations for improved performance outcomes.

¹¹<https://pywavelets.readthedocs.io/en/latest/index.html>

Table 5: Perplexity without Non-overlapping Inference. We evaluated the validation set in extrapolation experiments using Wikitext-103. Maximum allowable length of sequences in pre-training is $L_{\text{train}} = 512$.

| Wavelet type | Sequence Length | | | | | |
|-----------------------------|-----------------|--------------|--------------|--------------|--------------|--------------|
| | 128 | 256 | 512 | 1012 | 1512 | 2512 |
| Continuous Wavelet Families | | | | | | |
| Ricker | 23.64 | 20.82 | 19.19 | 18.23 | 18.00 | 17.99 |
| Gaussian | 23.77 | 20.90 | 19.30 | 18.31 | 18.02 | <u>17.88</u> |
| Morlet | 24.15 | 21.28 | 19.65 | 19.02 | 20.46 | 26.56 |
| Discrete Wavelet Families | | | | | | |
| Haar | 23.73 | 20.89 | 19.27 | 18.34 | 18.11 | 18.17 |
| db2 | 25.22 | 22.26 | 20.64 | 30.30 | 60.27 | 130.93 |
| db4 | 25.22 | 22.47 | 21.37 | 41.78 | 51.75 | 56.18 |
| db8 | 25.19 | 22.48 | 21.58 | 26.90 | 31.55 | 39.75 |
| db16 | 25.23 | 22.43 | 21.24 | 21.15 | 22.16 | 46.65 |
| db32 | 25.12 | 22.35 | 21.14 | 21.20 | 22.40 | 38.00 |
| sym2 | 25.11 | 22.21 | 20.68 | 31.25 | 61.00 | 126.32 |
| sym4 | 25.27 | 22.56 | 21.98 | 24.70 | 26.81 | 42.81 |
| sym8 | 29.27 | 26.13 | 24.63 | 23.97 | 31.47 | 92.36 |
| coif1 | 31.24 | 28.00 | 26.24 | 64.62 | 71.06 | 97.60 |
| coif2 | 25.24 | 22.47 | 21.39 | 27.74 | 27.39 | 44.26 |
| coif4 | 49.91 | 45.15 | 42.42 | 41.07 | 56.08 | 110.27 |
| coif8 | 25.15 | 22.39 | 21.26 | 21.31 | 22.26 | 35.73 |
| coif16 | 126.38 | 117.88 | 113.42 | 132.14 | 166.77 | 230.95 |
| dmey | 30.38 | 27.12 | 25.45 | 25.88 | 46.35 | 131.48 |
| bior1.3 | 26.27 | 23.36 | 23.69 | 23.38 | 30.71 | 88.66 |
| bior2.2 | 25.28 | 22.51 | 21.59 | 29.71 | 29.43 | 50.25 |
| bior2.6 | 25.29 | 22.70 | 21.60 | 22.15 | 22.71 | 40.61 |
| bior3.1 | 26.92 | 24.02 | 22.38 | 59.30 | 113.81 | 205.54 |
| bior3.5 | 25.17 | 22.49 | 21.65 | 27.41 | 27.19 | 53.99 |
| bior3.9 | 25.24 | 22.48 | 21.51 | 21.89 | 23.86 | 50.14 |
| bior4.4 | 25.52 | 22.72 | 21.64 | 21.67 | 24.42 | 51.46 |
| bior5.5 | 25.21 | 22.55 | 21.72 | 23.43 | 24.68 | 36.30 |
| bior6.8 | 25.14 | 22.39 | 21.21 | 21.10 | 22.31 | 46.97 |
| rbio1.1 | 24.26 | 21.34 | 19.69 | 18.79 | 18.63 | 18.98 |
| rbio1.3 | 25.28 | 22.50 | 21.39 | 52.06 | 47.78 | 59.94 |
| rbio2.2 | 25.92 | 23.08 | 21.98 | 68.57 | 86.12 | 93.90 |
| rbio2.6 | 25.29 | 22.68 | 21.60 | 24.54 | 24.47 | 44.57 |

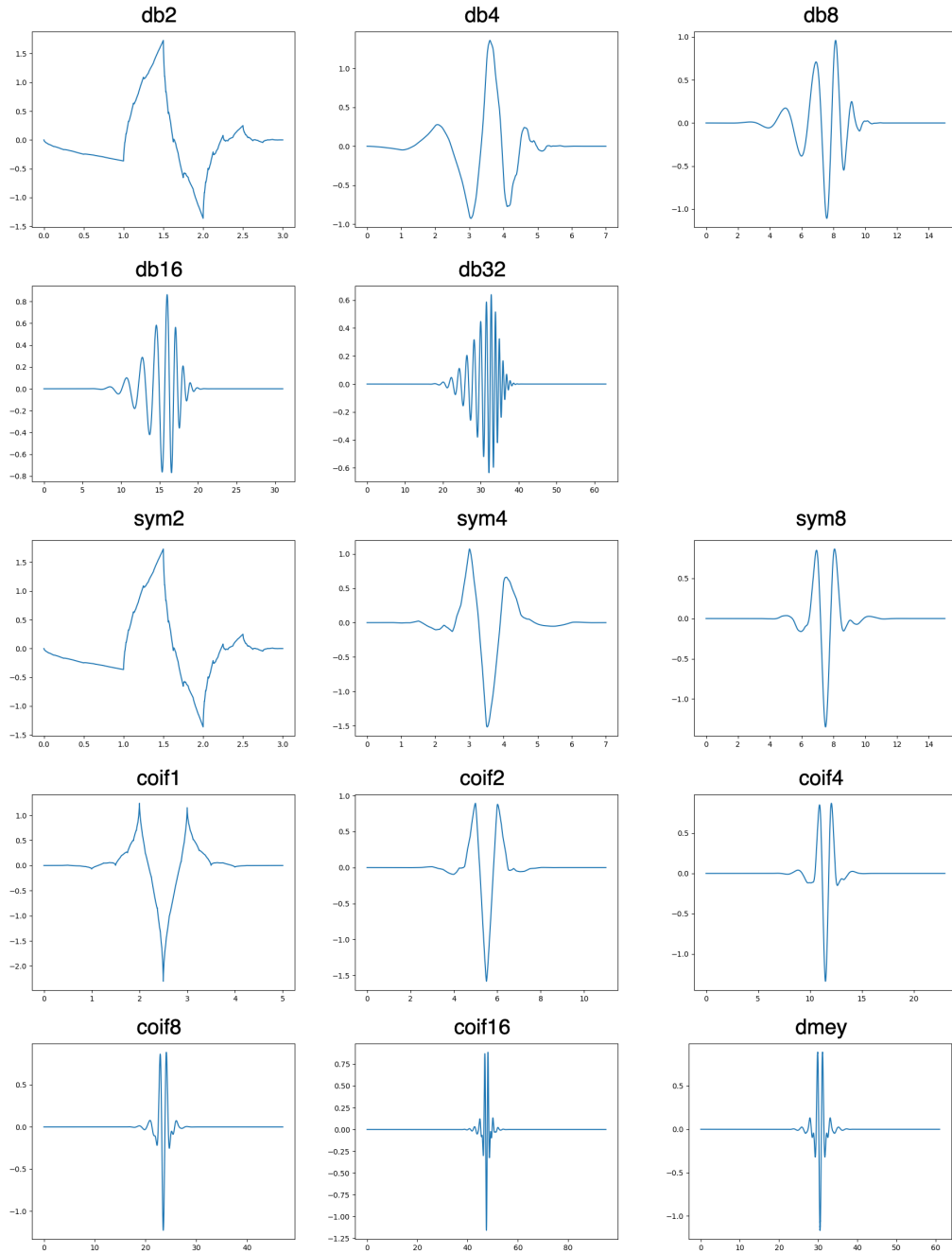


Figure 8: Graph of compared wavelet with level=10. Pywavelet (Lee et al., 2019) was used to calculate wavelets.

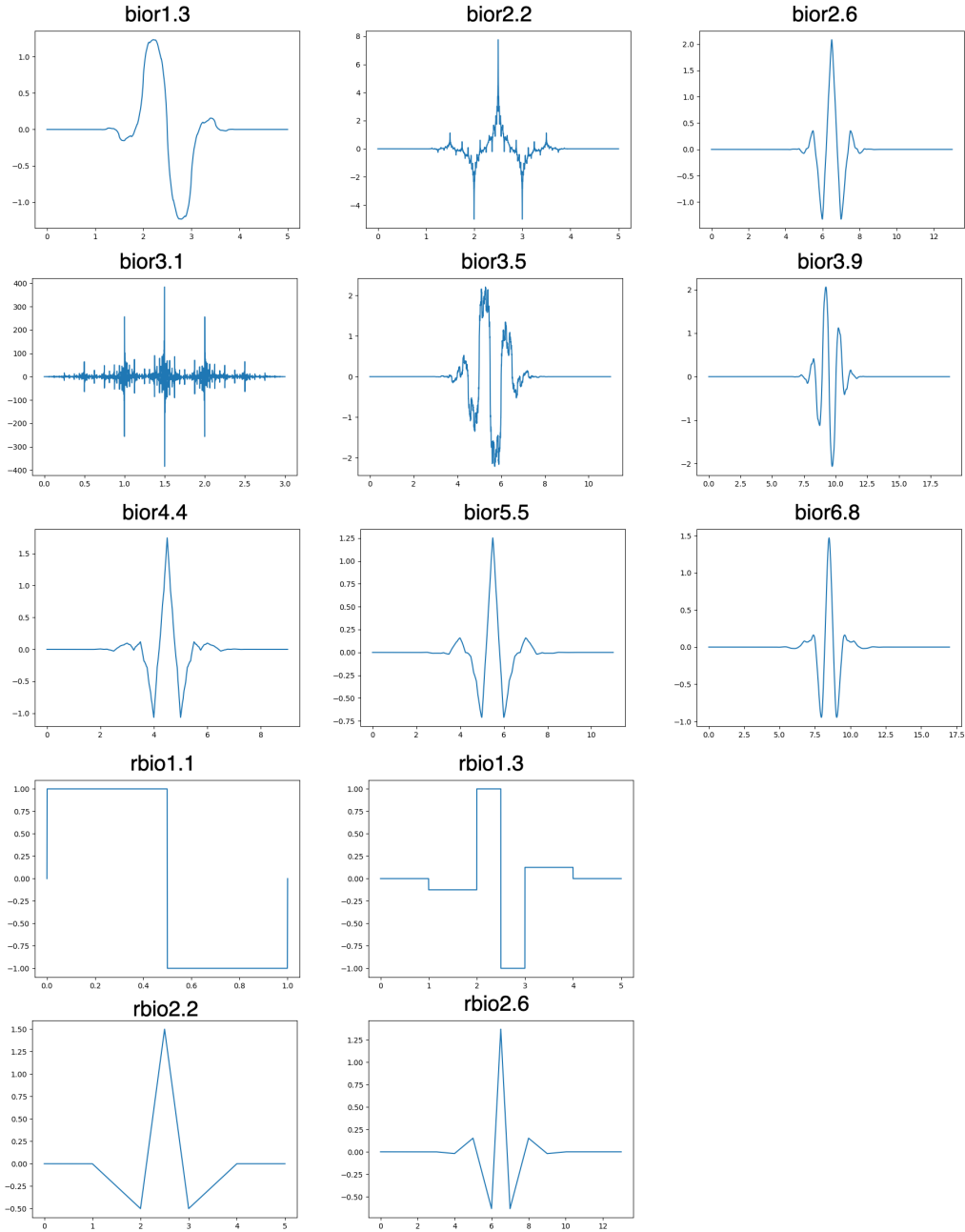


Figure 9: Graph of compared wavelet with level=10. Pywavelet (Lee et al., 2019) was used to calculate wavelets.

A.12 EXAMPLE OF HEAT MAP AND TEXT CORRESPONDENCE

Figure 10 shows the attention map after softmax operation for the proposed method. First, the notable feature of the proposed method is that it is always able to pay attention to specific tokens. The words that always receive attention are words that are important in the sentence, such as the '</s>' token, the first token, and words that are the subject of the sequence, such as 'he.' Moreover, as with ALiBi, the proposed method has a different scope of attention for each head.

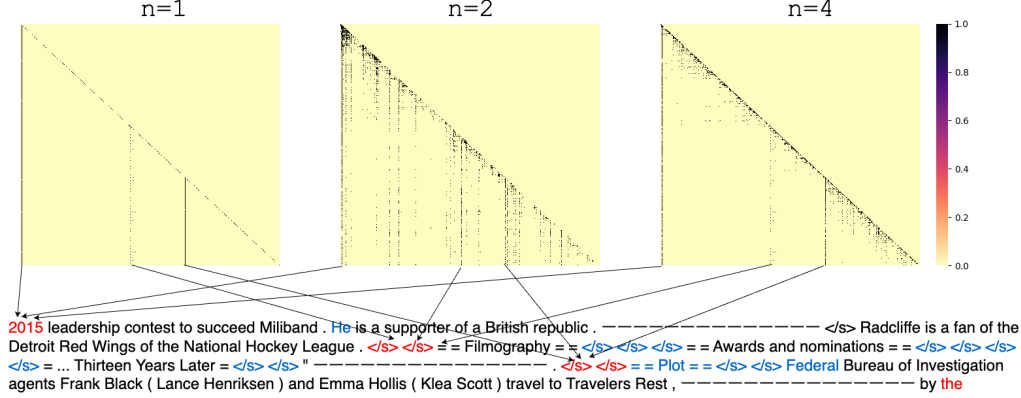


Figure 10: Heatmap of attention score e_{ij} after softmax operation for the proposed method. The maximum sequence length is $L_{max} = 512$ and the sequence length at inference is $L = 1012$. From left to right, $n = 1, 2, 4$ th heads are shown. Scores above 0.01 are mapped in black and the rest in yellow. Words that were always paid attention in all heads are shown in red, and words that were frequently paid attention only in the $n = 2$ th head are shown in blue. Sentences are omitted in the middle because they are long with 1012 tokens.