Exploring Transformer Extrapolation

Zhen Qin[1,2*], **Yiran Zhong**[1*†], **Hui Deng**[3]

[1]OpenNLPLab, Shanghai AI Lab, Shanghai, China
[2]TapTap, Shanghai, China
[3]Northwestern Polytechnical University, Shaanxi, China
{zhenqin950102, zhongyiran}@gmail.com, denghui986@foxmail.com

Abstract

Length extrapolation has attracted considerable attention recently since it allows transformers to be tested on longer sequences than those used in training. Previous research has shown that this property can be attained by using carefully designed Relative Positional Encodings (RPEs). While these methods perform well on a variety of corpora, the conditions for length extrapolation have yet to be investigated. This paper attempts to determine what types of RPEs allow for length extrapolation through a thorough mathematical and empirical analysis. We discover that a transformer is certain to possess this property as long as the series that corresponds to the RPE's exponential converges. Two practices are derived from the conditions and examined in language modeling tasks on a variety of corpora. As a bonus from the conditions, we derive a new Theoretical Receptive Field (TRF) to measure the receptive field of RPEs without taking any training steps. Extensive experiments are conducted on the Wikitext-103, Books, Github, and WikiBook datasets to demonstrate the viability of our discovered conditions. We also compare TRF to Empirical Receptive Field (ERF) across different models, showing consistently matched trends on these datasets. Code is released at: https://github.com/OpenNLPLab/Rpe.

Introduction

Transformer (Vaswani et al. 2017) is advancing steadily in the areas of natural language processing (Qin et al. 2023b; Devlin et al. 2019; Liu et al. 2019; Qin et al. 2022b,a; Liu et al. 2022; Qin and Zhong 2023), computer vision (Dosovitskiy et al. 2020; Sun et al. 2022b; Lu et al. 2022; Hao et al. 2024), and audio processing (Gong, Chung, and Glass 2021; Akbari et al. 2021; Gulati et al. 2020; Sun et al. 2022a). Although it outperforms other architectures such as RNNs (Cho et al. 2014; Qin, Yang, and Zhong 2023) and CNNs (Kim 2014; Hershey et al. 2016; Gehring et al. 2017) in many sequence modeling tasks, its lack of length extrapolation capability limits its ability to handle a wide range of sequence lengths, *i.e.,* inference sequences need to be equal to or shorter than training sequences. Increasing the training sequence length is only a temporary solution because the space-time complexity grows quadratically with the sequence length. Another option is to extend the inference sequence length by converting the trained full attention blocks to sliding window attention blocks (Beltagy, Peters, and Cohan 2020), but this will result in significantly worse efficiency than the full attention speed (Press, Smith, and Lewis 2022). How to permanently resolve this issue without incurring additional costs has emerged as a new topic.

A mainstream solution for length extrapolation is to design a Relative Positional Encoding (RPE) (Qin et al. 2023c) that concentrates attention on neighboring tokens. For example, ALiBi (Press, Smith, and Lewis 2022) applies linear decay biases to the attention to reduce the contribution from distant tokens. Kerple (Chi et al. 2022) investigates shift-invariant conditionally positive definite kernels in RPEs and proposes a collection of kernels that promote the length extrapolation property. It also shows that ALiBi is one of its instances. Sandwich (Chi, Fan, and Rudnicky 2022) proposes a hypothesis to explain the secret behind ALiBi and empirically proves it by integrating the hypothesis into sinusoidal positional embeddings.

In order to investigate transformer extrapolation, we first establish a hypothesis regarding why existing RPE-based length extrapolation methods (Qin et al. 2023a) have this capacity to extrapolate sequences in inference based on empirical analysis. Then we identify the conditions of RPEs that satisfy the hypothesis through mathematical analysis. Finally, the discovered conditions are empirically validated on a variety of corpora. Specifically, we assume that due to decay biases, existing RPE-based length extrapolation methods behave similarly to sliding window attention, *i.e.,* only tokens within a certain range can influence the attention scores. A transformer can extrapolate for certain in this scenario since the out-of-range tokens have no effect on the attention outcomes. We derive that a transformer is guaranteed to satisfy this hypothesis if the series corresponding to the exponential of its RPE converges. Based on the observation, we show that previous RPE-based methods (Press, Smith, and Lewis 2022; Chi et al. 2022) can be seen as particular instances under the conditions. Two new practices from the conditions are derived and evaluated in language modeling.

The observed conditions not only shed light on the secret of length extrapolation but also offer a new perspective on computing the Theoretical Receptive Fields (TRF) of RPEs. In contrast to prior approaches that require training gradients

[*] These authors contributed equally.

[†] Corresponding author.

to compute TRF, we propose a new way to calculate TRF that is solely based on the formulation of RPEs. Extensive experiments on various datasets validate the conditions. TRF calculated by our method substantially matches the trend of the Empirical Receptive Field (ERF) in real-world scenarios.

Preliminary

Before embarking on the journey of exploring, we introduce several preliminary concepts that will be used throughout the paper, such as softmax attention, relative positional encoding, length extrapolation, and sliding window attention. We also provide the necessary notations for the subsequent analysis, *i.e.,* we use $\mathbf{M}$ to denote a matrix and $\mathbf{m}_i^\top$ to represent the ith row of $\mathbf{M}$. The complete math notations can be found in Appendix. Following previous work (Press, Smith, and Lewis 2022), we restrict our analysis to causal language models and assume that the max sequence length during training is m.

Softmax Attention

Softmax attention is a key component of transformers which operates on query $\mathbf{Q}$, key $\mathbf{K}$ and value $\mathbf{V}$ matrices. Each matrix is a linear map that takes $\mathbf{X} \in \mathbb{R}^{n\times d}$ as input:

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_Q,\ \mathbf{K} = \mathbf{X}\mathbf{W}_K,\ \mathbf{V} = \mathbf{X}\mathbf{W}_V \in \mathbb{R}^{n\times d}, \tag{1}$$

where n is the sequence length and d is the dimension of the hidden feature. The output attention matrix $\mathbf{O} \in \mathbb{R}^{n\times d}$ can be formulated as:

$$\mathbf{O} = \mathrm{Softmax}(\mathbf{Q}\mathbf{K}^\mathsf{T}/\sqrt{d})\mathbf{V}. \tag{2}$$

To prevent information leakage in causal language modeling, a mask matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ is used to ensure that current tokens can only see previous tokens and themselves. The lower triangular elements of $\mathbf{M}$ are 0, and the upper triangular ones, except for the diagonal, are $-\infty$. Then the output attention matrix $\mathbf{O}$ for causal language models will be:

$$\mathbf{O} = \mathrm{Softmax}(\mathbf{Q}\mathbf{K}^\top/\sqrt{d} + \mathbf{M})\mathbf{V}. \tag{3}$$

Note that Eq. 3 can be seen as a general form of attention, *i.e.,* when the elements of $\mathbf{M}$ are all 0, Eq. 3 is degenerated to Eq. 2. For ease of discussion, we use Eq. 3 to represent attention computation.

Relative Positional Encoding

Positional encoding is designed to inject positional bias into transformers. Absolute Positional Encoding (APE) (Vaswani et al. 2017; Gehring et al. 2017) and Relative Positional Encoding (RPE) (Su et al. 2021; Liutkus et al. 2021; Press, Smith, and Lewis 2022; Chi et al. 2022) are the two most common types of positional encoding. In this paper, we focus on RPE because it is the key for length extrapolation, as shown in (Press, Smith, and Lewis 2022). An attention with RPE can be written as:

$$\mathbf{O} = \mathrm{Softmax}(\mathbf{Q}\mathbf{K}^\top/\sqrt{d} + \mathbf{M} + \mathbf{P})\mathbf{V}, \tag{4}$$

where $\mathbf{P} \in \mathbb{R}^{n\times n}$ is a Toeplitz matrix that encodes relative positional information, *i.e.,* $p_{ij} = p_{i-j}$. It is worth noting that $\mathbf{M}$ and $\mathbf{P}$ can be merged, and the merged matrix is still a Toeplitz matrix. We use $\mathbf{R}$ to represent the merged matrix and rewrite Eq. 4 as:

$$\mathbf{O} = \mathrm{Softmax}(\mathbf{Q}\mathbf{K}^\top/\sqrt{d} + \mathbf{R})\mathbf{V}. \tag{5}$$

Definition Of Length Extrapolation

The property of length extrapolation allows a model to be tested on longer sequences than those used in training. Previous sequence modeling structures such as RNNs (Hochreiter and Schmidhuber 1997) and CNNs (Gehring et al. 2017) often naturally possess this property, but it is a difficult task for transformers. This property is only present in sliding window transformers and a few transformer variants with specifically designed RPEs (Chi et al. 2022; Press, Smith, and Lewis 2022; Chi, Fan, and Rudnicky 2022). In language modeling, one token can only see itself and previous tokens. Therefore, regardless the sequence length, the performance should be stable for the neighboring tokens that are within the training sequence length (Beltagy, Peters, and Cohan 2020). For the tokens that are out of range, the performance will degrade if the model does not support length extrapolation (Press, Smith, and Lewis 2022). Based on the observation above, we give a definition of length extrapolation:

Definition 0.1. *For a language model $\mathcal{F}$, given dataset $\mathcal{X}$, if for any n, there is,*

$$|ppl_n(\mathcal{X},\mathcal{F}) - ppl_m(\mathcal{X},\mathcal{F})|/ppl_m(\mathcal{X},\mathcal{F}) < \delta, \tag{6}$$

then $\mathcal{F}$ is considered to have the extrapolation property.

Here $\delta > 0$ is a small constant, $ppl_n(\mathcal{X},\mathcal{F})$ means that $\mathcal{F}$ calculates perplexity with a max sequence length of n on the data set $\mathcal{X}$. Empirically, if $|ppl_n(\mathcal{X},\mathcal{F}) - ppl_m(\mathcal{X},\mathcal{F})|/ppl_m(\mathcal{X},\mathcal{F})$ becomes very large($\gg 1$) as n increases, we consider that $\mathcal{F}$ does not have extrapolation property.

Sliding Window Attention

For the convenience of subsequent discussions, we define a window attention at position i and window size j as follows:

$$\begin{aligned}\mathbf{o}_i^j &= \frac{\sum_{i-j+1\le s\le i}\exp(\mathbf{q}_i^\top\mathbf{k}_s/\sqrt{d})\exp(r_{is})\mathbf{v}_s}{\sum_{i-j+1\le t\le i}\exp(\mathbf{q}_i^\top\mathbf{k}_t/\sqrt{d})\exp(r_{it})}\\ &\triangleq \frac{\sum_{i-j+1\le s\le i} c_{is}\mathbf{v}_s}{C_{ij}},\end{aligned} \tag{7}$$

where $C_{ij} = \sum_{i-j+1\le t\le i} c_{it}, c_{ij} = a_{ij}b_{ij}, a_{ij} = \exp(\mathbf{q}_i^\top\mathbf{k}_j/\sqrt{d}), b_{ij} = \exp(r_{ij}), j \le i.$

We further assume $\|\mathbf{x}_i\| \le l, \mathbf{x} \in \{\mathbf{q},\mathbf{k},\mathbf{v}\}$, where $l > 0$ is a constant. The $\mathbf{o}_i^j$ represents the attention output of the i-th token, which interacts with the j tokens preceding it. Note that window attention naturally possesses the length extrapolation ability.

There are two ways to infer window attention: nonoverlapping inference and sliding window inference as shown on the right of Figure 1. In sliding window inference, the tokens within each sliding window must be re-encoded multiple times, making it substantially slower than the nonoverlapping one. In Table 1 we compare the average inference time over a group of window sizes between the sliding window inference and nonoverlapping window inference. The sliding window one is more than 44 times slower than the nonoverlapping one. However, as shown on the left of Figure 1, the sliding window inference has much lower ppl than the nonoverlapping one.

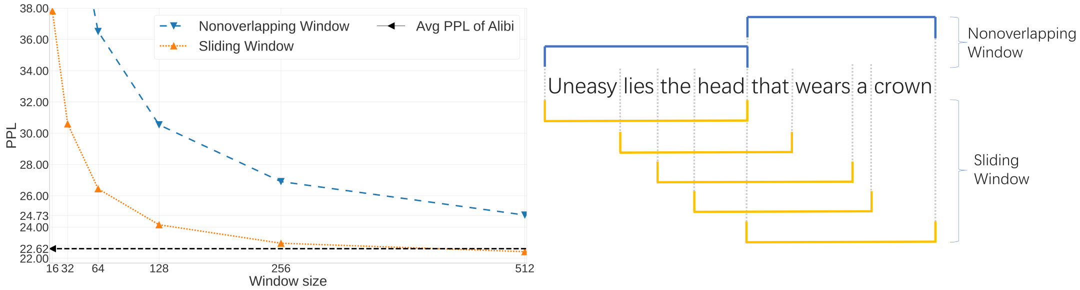
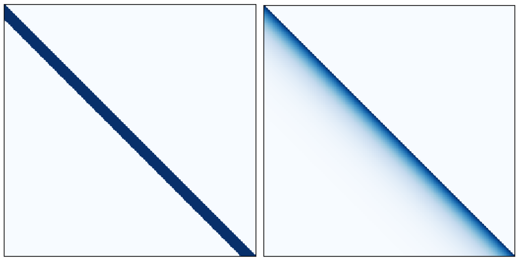


Figure 1: Sliding window inference *vs* Nonoverlapping inference. We illustrate the difference between sliding window inference and nonoverlapping inference in the right figure. The left figure shows the curves of ”Sliding Window“ and ”Nonoverlapping Window“corresponding to the ppls calculated by a language model at different inference window sizes.

Method	Rel Avg infer time
Sliding Window	44.35
Nonoverlapping Window	1.00
Alibi	1.00

Table 1: Relative average inference time. We compute the relative average inference time of sliding window inference and nonoverlapping inference over a set of window sizes $\{16,32,64,128,258,512\}$. We also include the Alibi inference time as a reference.



(a) Sliding window (b)Alibi linear decay

Figure 2: Visualization of attention reweighting. We plot the reweighting schema of sliding window attention and Alibi linear decay bias. They share a similar behavior in that only neighboring tokens can influence the attention results.

Transformer Extrapolation Exploration

In this section, we first describe the hypothesis about why existing RPE-based length extrapolation methods can extrapolate sequences in inference and provide empirical evidence for it. Then we derive the conditions for length extrapolation in detail and demonstrate that recent RPE-based length extrapolation methods (Chi et al. 2022; Press, Smith, and Lewis 2022) satisfy the conditions.

The Hypothesis

A sliding window attention with window size w is equivalent to the following RPE on full attention:

$$\mathrm{m}_{ij} = \begin{cases} 0, & i-j \le w. \\ -\infty, & \text{others}. \end{cases} \tag{8}$$

By comparing Eq. 8 and the corresponding RPE of Alibi (Press, Smith, and Lewis 2022) in Figure 2, we can see that they both have the same behavior in that they both concentrate tokens inside a specified range. Also, in Figure 1, we show that the performance of Alibi is similar to the sliding window attention when the window size is sufficiently large. Based on these two observations, we make the following hypothesis:

Hypothesis 0.1. *A RPE that makes a transformer extrapolatable needs to have similar behavior to sliding window attention,* i.e., $\delta(i,j)$ *should satisfy:*

$$\forall \epsilon > 0, \exists j_0, s.t, j > j_0, \delta(i,j) < \epsilon, \tag{9}$$

where $\delta(i,j) \triangleq \|\mathbf{o}_i^i - \mathbf{o}_i^j\|$, *and the window length* j *needs to be sufficiently large.*

In the following sections, we will derive the conditions for RPEs that satisfy Eq. 9.

The Conditions

Let us introduce the first lemma:

Lemma 0.2. *When the following condition is satisfied, Eq. 9 holds.*

$$\lim_{i\to\infty} C_{ii} \triangleq C < \infty. \tag{10}$$

Proof. When $i \le m$, the test sequence length is smaller than the max sequence length m during training, take $j = i$, we get $\|\mathbf{o}_i^i - \mathbf{o}_i^j\| = \|\mathbf{o}_i^i - \mathbf{o}_i^i\| = 0$. When $i > m$, we can reformulate Eq. 7 as:

$$\begin{aligned} \mathbf{o}_i^i &= \frac{\sum_{i-j+1\le s\le i} c_{is}\mathbf{v}_s + \sum_{1\le s\le i-j} c_{is}\mathbf{v}_s}{C_{ii}} \\ &= \frac{\sum_{i-j+1\le s\le i} c_{is}\mathbf{v}_s}{C_{ij}} \frac{C_{ij}}{C_{ii}} + \frac{\sum_{1\le s\le i-j} c_{is}\mathbf{v}_s}{C_{ii}-C_{ij}} \frac{C_{ii}-C_{ij}}{C_{ii}} \\ &= \frac{\sum_{i-j+1\le s\le i} c_{is}\mathbf{v}_s}{C_{ij}} \frac{C_{ij}}{C_{ii}} + \frac{\sum_{1\le s\le i-j} c_{is}\mathbf{v}_s}{C_{ii}-C_{ij}} \left(1 - \frac{C_{ij}}{C_{ii}}\right). \end{aligned}$$

Therefore we have $\mathbf{o}_i^i - \mathbf{o}_i^j =:$

$$\left(1 - \frac{C_{ij}}{C_{ii}}\right)\left(\frac{\sum_{i-j+1\le s\le i} c_{is}\mathbf{v}_s}{C_{ij}} - \frac{\sum_{1\le s\le i-j} c_{is}\mathbf{v}_s}{C_{ii}-C_{ij}}\right). \tag{11}$$

For the second part:

$$\begin{aligned}&\left\|\frac{\sum_{i-j+1\le s\le i} c_{is}\mathbf{v}_s}{C_{ij}} - \frac{\sum_{1\le s\le i-j} c_{is}\mathbf{v}_s}{C_{ii}-C_{ij}}\right\| \\ &\le \frac{\sum_{i-j+1\le s\le i} c_{is}\|\mathbf{v}_s\|}{C_{ij}} + \frac{\sum_{1\le s\le i-j} c_{is}\|\mathbf{v}_s\|}{C_{ii}-C_{ij}} \\ &\le \frac{\sum_{i-j+1\le s\le i} c_{is}l}{C_{ij}} + \frac{\sum_{1\le s\le i-j} c_{is}l}{C_{ii}-C_{ij}} = 2l\end{aligned} \tag{12}$$

We have

$$\delta(i,j) \le 2\left(1-\frac{C_{ij}}{C_{ii}}\right)l. \tag{13}$$

According to Eq 10 and the tail of convergence series is arbitrarily small. $\forall C/2 > \epsilon > 0$, we can find a j_0, s.t. if $i \ge j > j_0$, $C_{ii} - C_{ij} < \epsilon$. We can also find a j_1, s.t. if $i \ge j > j_1$, $C - \epsilon < C_{ii} < C + \epsilon$. If we take $j_2 = \max(j_0, j_1)$, so if $i \ge j \ge j_2$, we have:

$$C_{ii} - C_{ij} < \epsilon, C - \epsilon < C_{ii} < C + \epsilon \tag{14}$$

So when $i \ge j \ge j_2$, we have

$$\begin{aligned}\delta(i,j) \le 2\left(1-\frac{C_{ij}}{C_{ii}}\right)l = 2\frac{C_{ii}-C_{ij}}{C_{ii}}l \le 2\frac{\epsilon}{C-\epsilon}l \\ \le \frac{2l\epsilon}{C-C/2} = \frac{4l\epsilon}{C}\end{aligned} \tag{15}$$

According to the definition of limitation, Eq. 10 holds. □

This lemma implies that for any token if the attention of the model focuses on its neighboring $j (j \ge j_2)$ tokens, the model has length extrapolation property. The lemma accompanies our intuitions. Does it mean that as long as a RPE follows the same principle, *i.e.,* places more weights on neighboring j tokens, the model is guaranteed to have the length extrapolation property? In the following sections, we will demonstrate that concentrating more weights on neighboring tokens does not guarantee the transformer has the length extrapolation property. Specifically, we will provide a mathematical proof of the sufficient conditions for RPE to have the length extrapolation property.

Theorem 0.3. *When the following condition is satisfied, Eq. 9 holds.*

$$\lim\nolimits_{i\to\infty} B_{ii} < \infty, B_{ii} = \sum_{1\le t\le i} b_{it} < \infty. \tag{16}$$

Proof. Since we assume $\|\mathbf{q}_i\| \le l, \|\mathbf{k}_i\| \le l$, then:

$$a_{ij} = \exp(\mathbf{q}_i^\top \mathbf{k}_j) \le \exp(l^2), \tag{17}$$

$$c_{ij} = a_{ij}b_{ij} \le \exp(l^2)b_{ij}, C_{ii} \le \exp(l^2)B_{ii}. \tag{18}$$

Therefore, Eq. 10 can be derived from Eq. 16. Combine with Lemma 0.2, the proof is concluded. □

By leveraging the property of RPE, Theorem 0.3 can be further simplified as:

Theorem 0.4. *When the following condition is satisfied, Eq. 9 holds.*

$$\lim_{i\to\infty}\sum_{t=1}^{i} b_{i-t} = \lim_{i\to\infty}\sum_{t=0}^{i-1} b_t < \infty. \tag{19}$$

Proof. According to the definition of RPE:

$$B_{ii} = \sum_{1\le t\le i} b_{it} = \sum_{t=1}^{i} b_{i-t} = \sum_{t=0}^{i-1} b_t. \tag{20}$$

This means that Eq. 16 is equivalent to:

$$\lim_{i\to\infty} B_{ii} = \lim_{i\to\infty}\sum_{t=0}^{i-1} b_t < \infty. \tag{21}$$

□

Theorem 0.4 indicates that as long as the series of $\exp(\text{RPE})$ converges, the model is guaranteed to have length extrapolation property. Based on this principle, we can mathematically determine whether an RPE allows for length extrapolation before conducting experiments or designing a variety of RPEs that can do length extrapolation. In Appendix, we show that previous methods such as Alibi (Press, Smith, and Lewis 2022), Kerple (Chi et al. 2022), and Sandwich (Chi, Fan, and Rudnicky 2022) satisfy our derived conditions for length extrapolation.

Theoretical Receptive Field

In the previous section, we established the conditions for length extrapolation. As an extra bonus, we can derive Theoretical Receptive Fields (TRF) for any RPE-based length extrapolation method. Let us start with the definition of Empirical Receptive Field (ERF). ERF can be viewed as a window containing the vast majority of the information contained within the attention.

Recall Eq. 13, by setting $1 - \frac{C_{ij}}{C_{ii}} = \epsilon$, we can define:

$$C_{ij} = C_{ii}(1-\epsilon),\ n_{\text{emp}}(\epsilon) = \inf_j (C_{ij} > C_{ii}(1-\epsilon)),$$

$n_{\text{emp}}(\epsilon)$ is the ERF that represents the minimal sequence length required to maintain the performance within a gap of ϵ. Intuitively, ERF can be viewed as the smallest window that contains the majority of the information within an attention. Since it is related to both a_{ij} and b_{ij}, it can only be calculated after training.

Now we define TRF, which allows us to estimate the receptive field without training. To accomplish this, we consider the upper bound of C_{ij}. From the definition of C_{ij} and Eq. 17, C_{ij} is upper bounded by B_{ij}. Therefore, we can define the TRF $n^b_{\text{the}}(\epsilon)$ respect to series b_t as:

$$\begin{aligned} n_{\text{the}}(\epsilon) &= \inf_j (B_{ij} > B(1-\epsilon)) \\ &= \inf_j \left(\sum_{t=0}^{j-1} b_t > B(1-\epsilon)\right) \\ &= \inf_j \left(\sum_{t\ge j} b_t < B\epsilon\right) \end{aligned} \tag{22}$$

where $B = \lim_{j\to\infty}\sum_{t=0}^{j-1} b_t$. We may find it difficult to give the analytical form of the partial sum of the series at times, but we can still compute the TRF numerically or compare the TRFs of different RPEs using the theorem below:

Theorem 0.5. *If the following conditions hold:*

$$\frac{\alpha_t}{\alpha} \le \frac{\beta_t}{\beta}, t \to \infty,\ \alpha \triangleq \lim_{j\to\infty}\sum_{t=0}^{j-1}\alpha_t,\ \beta \triangleq \lim_{j\to\infty}\sum_{t=0}^{j-1}\beta_t. \tag{23}$$

Then:

$$n^{\alpha}_{\mathrm{the}}(\epsilon) \le n^{\beta}_{\mathrm{the}}(\epsilon), \epsilon \to 0. \tag{24}$$

Proof. According to Eq.23, there exists $t_0 > 0$, such that, when $t > t_0$, we have:

$$\frac{\alpha_t}{\alpha} \le \frac{\beta_t}{\beta}. \tag{25}$$

Let $\epsilon < \epsilon_0$, where

$$n^{\beta}_{\mathrm{the}}(\epsilon_0) = t_0, \tag{26}$$

then we get:

$$\sum_{t \ge n^{\beta}_{\mathrm{the}}(\epsilon)} \beta_t \le \beta\epsilon, n^{\beta}_{\mathrm{the}}(\epsilon) > t_0. \tag{27}$$

Finally:

$$\sum_{t \ge n^{\beta}_{\mathrm{the}}(\epsilon)} \alpha_t \le \sum_{t \ge n^{\beta}_{\mathrm{the}}(\epsilon)} \frac{\alpha\beta_t}{\beta} \le \frac{\alpha\beta\epsilon}{\beta} = \alpha\epsilon.$$

According to Eq. 22, we have:

$$n^{a}_{\mathrm{the}}(\epsilon) \le n^{b}_{\mathrm{the}}(\epsilon). \tag{28}$$

The $\exp(\mathrm{RPE})$ series follows the same trend as TRF, the smaller the series, the smaller the TRF. □

We provide several examples of how to compute TRF in the Appendix.

Two New RPEs

Based on the proven conditions of length extrapolation, we can design infinite kinds of RPEs with the length extrapolation property. Here, we propose two new RPEs to empirically prove the conditions and hypothesis, namely:

$$\begin{aligned}\mathrm{Type1}: b_n &= \frac{1}{n^2} = \exp(-2\ln n),\\ \mathrm{Type2}: b_n &= \exp(-\ln^2 n);\end{aligned}$$

The corresponding TRF of Type 1 is:

$$\begin{aligned} B_{ij} &= \sum_{i=0}^{j-1}\frac{1}{(i+1)^2} \approx \int_1^j \frac{1}{x^2}dx = 1 - \frac{1}{j}, B = 1.\\ n_{\mathrm{the}}(\epsilon) &= \inf_j\left(B_{ij} > B(1-\epsilon)\right) \\ &= \inf_j\left(1 - \frac{1}{j} > 1 - \epsilon\right) = \Theta\left(\frac{1}{\epsilon}\right) \end{aligned} \tag{29}$$

For Type 2, it is difficult to provide the analytical form of its TRF. However, we can prove that the TRF of Type 2 is smaller than the TRF of Type 1 using Theorem 0.5 and the inequality below:

$$\forall c_1, c_2 > 0, \frac{\exp(-\ln^2 n)}{c_1} < \frac{1/n^2}{c_2}, n \to \infty.$$

Empirical Validation

Setting All models are implemented in Fairseq (Ott et al. 2019) and trained on 8 V100 GPUs. We use the same model architecture and training configuration for all RPE variants to ensure fairness. For Wikitext-103 (Merity et al. 2016), since it is a relatively small dataset, we use a 6-layer transformer decoder structure with an embedding size of 512. For other datasets, in particular, we used a 12-layer transformer decoder structure with an embedding size of 768. The evaluation metric is perplexity (PPL) and the max training length during training is 512. The detailed hyper-parameter settings are listed in Appendix.

Dataset We conduct experiments on Wikitext-103 (Merity et al. 2016), Books (Zhu et al. 2015), Github (Gao et al. 2020) and WikiBook (Wettig et al. 2022). Wikitext-103 is a small dataset containing a preprocessed version of the Wikipedia dataset. It is widely used in many NLP papers. Books has a large number of novels, making it a good corpus for long sequence processing. Github consists of a sizable amount of open-source repositories, the majority of which are written in coding languages. WikiBook is a 22-gigabyte corpus of Wikipedia articles and books curated by (Wettig et al. 2022). This corpus is used to validate the performance of various models on large datasets.

Validating The Sufficiency. To empirically validate the sufficiency of our discovered conditions, we integrate the two RPEs that were proposed in the previous section into transformers and test their length extrapolation capability on Wikitext-103, Books, Github, and WikiBook datasets. We increase the length of the inference sequence from 512 to 9216 tokens and plot the testing PPLs of our proposed RPEs as well as those of existing methods such as Alibi, Kerple, and Sandwich in Figure 3. All these methods demonstrate good length extrapolation capability. However, the stabilized PPL may vary due to the effectiveness of different positional encoding strategies, which are not considered in this paper. We include the Sinusoidal (Vaswani et al. 2017) positional encoding as a reference method that cannot extrapolate, which grows rapidly as the inference sequence length increases.

Validating The Necessity. Although we only provide mathematical proof for the sufficiency of our discovered conditions, we also attempt to verify their necessity empirically in this section. Specifically, we pick two RPEs that are very close to satisfying Theorem 0.4 as follows. Note that both of them concentrate their weight on neighboring tokens.

$$\mathrm{Example1}: b_n = \frac{1}{n},\ \mathrm{Example2}: b_n = \frac{1}{n\ln n}$$

Below is a brief mathematical proof that the above RPEs do not satisfy Theorem 0.4.

$$\sum_{n=1}^{k}\frac{1}{n} > \int_1^{k+1}\frac{1}{x}dx = \ln(k+1),$$

$$\sum_{n=3}^{k}\frac{1}{n\ln n} > \int_3^{k+1}\frac{1}{x\ln x}dx = \ln\ln(k+1) - \ln\ln 3.$$

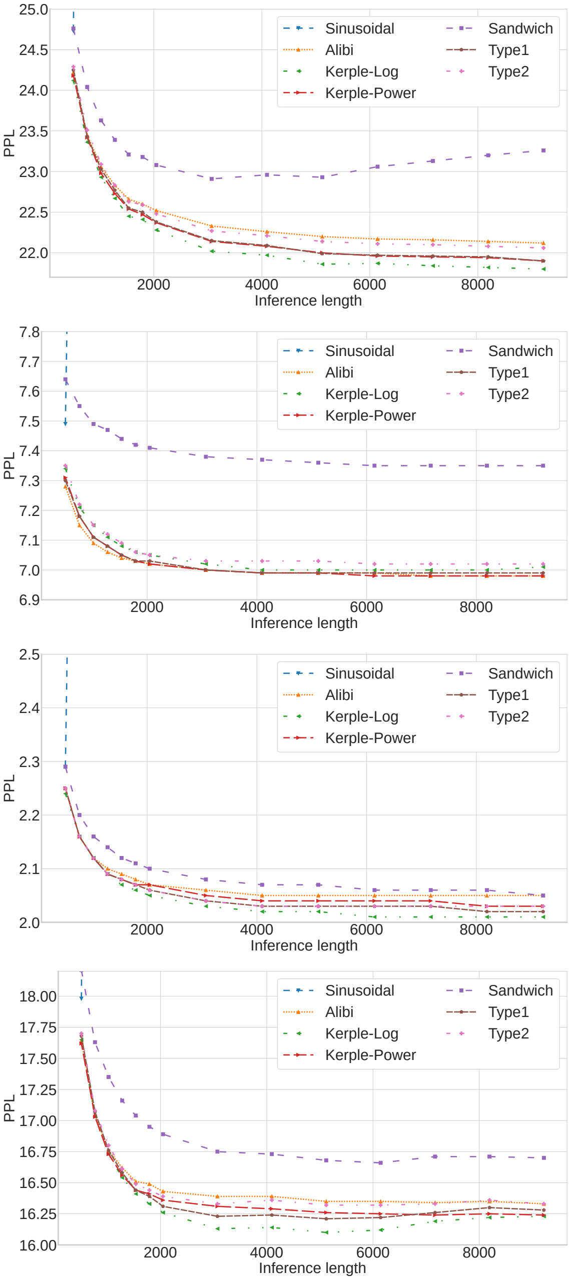


Figure 3: Sufficiency validation on Wikitext-103, Books, Github, WikiBook datasets (in top to down order). To test length extrapolation capability, we lengthen inference sequences from 512 to 9216 tokens and plot the testing PPLs of our proposed Type 1 and Type 2 RPEs, as well as Alibi, Kerple, and Sandwich.

We then empirically test their length extrapolation capability on Wikitext-103, Books, Github, and WikiBook datasets by scaling the inference sequence length from 512 to 9216 tokens. As shown in Figure 4, the PPLs of both RPEs grow rapidly as the length of the testing sequence increases. It demonstrates that both of them cannot extrapolate. We also include Type 1 RPE in Figure 4 as a reference.

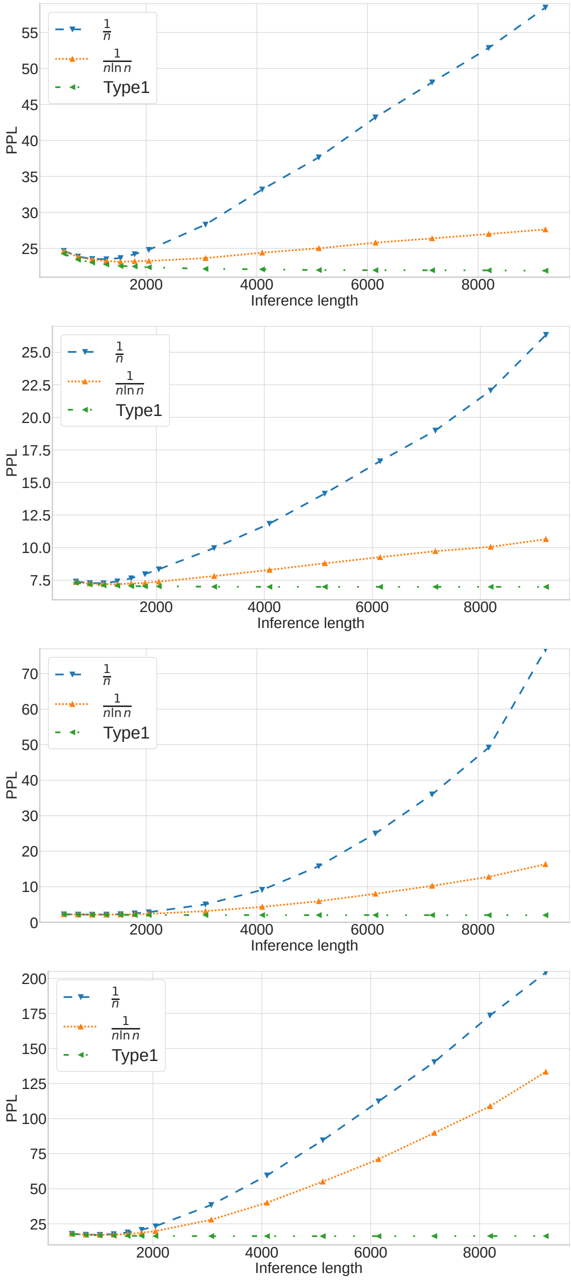


Figure 4: Necessity validation on Wikitext-103, Books, Github, WikiBook datasets (in top to down order). We select two RPEs that do not satisfying Theorem 0.4, *e.g.,* $b_n = \frac{1}{n}$ and $b_n = \frac{1}{n \ln n}$.

Validating TRF We validate our proposed TRF by comparing the trend between the TRF and ERF. We plot the TRFs and ERFs of the Alibi, Kerple, Sandwich, and our proposed RPEs on the aforementioned datasets. As observed in Figure 6 and Figure 5, while the curves vary across datasets, TRF estimates a similar overall trend of ERFs.

Visualizing RPE We visualize the weighting schemes of Type 1 and 2 in Figure 7, *i.e.,* the heatmap of $\exp(\text{RPE})$.

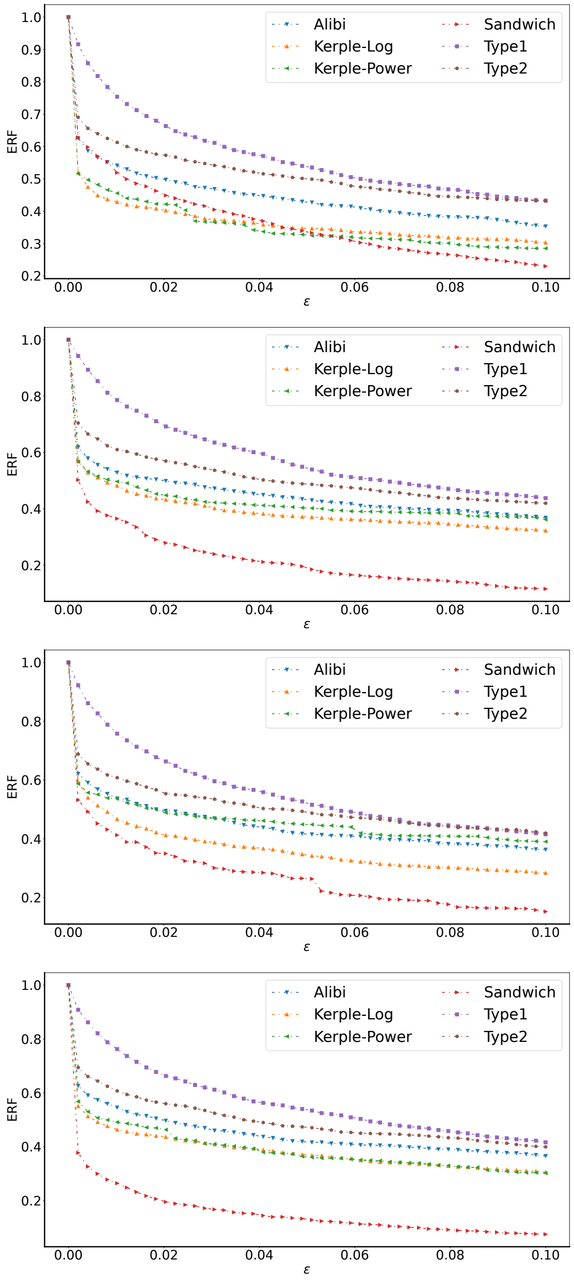


Figure 5: We plot the ERF for Alibi, Kerple, Sandwich and our proposed Type 1 and Type 2 methods on Wikitext-103, Books, Github, and WikiBook datasets using trained models. ERF is normalized for better visualization.

Type 2 concentrates weights on closer neighboring tokens than Type 1, indicating a smaller TRF and ERF as shown in Figure 6 and Figure 5. We also visualize other methods in Appendix.

Conclusion

In this paper, we explore the secrets of transformer length extrapolation in language modeling. We first make a hypothesis about extrapolation and then derived the sufficient conditions for RPE to have the length extrapolation property. A thorough mathematical analysis reveals that a transformer model is certain to be capable of length extrapolation if the series that corresponds to the exponential of its RPE converges. This observation brings an extra bonus: we can estimate TRFs of RPEs solely based on their formulations. We chose two new RPEs that satisfy the conditions and two that do not to empirically prove the sufficiency of the conditions on four widely used datasets. We also validated our TRFs by comparing them with ERFs on these datasets as well. The results show that our TRFs can accurately reflect the actual receptive fields of RPEs before training.

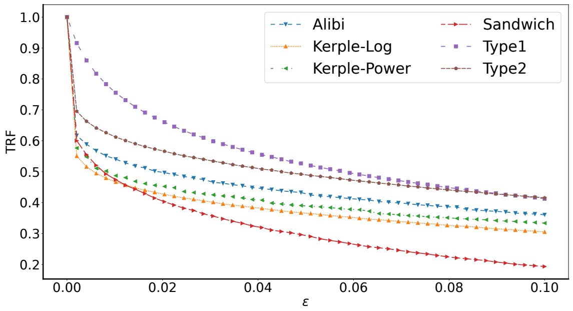
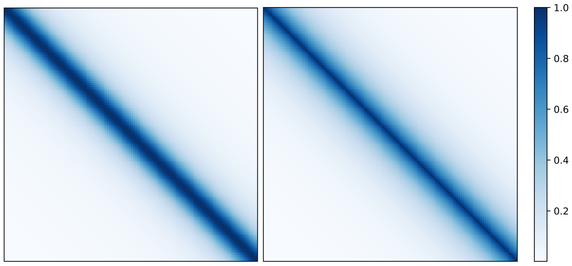


Figure 6: We numerically plot TRFs for existing methods and our proposed method. TRF is normalized for visualization. The TRF of Type 1 is larger than Type 2, which matches the Theorem 0.5 and our analysis.



(a) Type1 (b) Type2

Figure 7: We plot the heatmap of $\exp(\text{RPE})$ for Type 1 and Type 2. Type 2 concentrates weights on closer neighboring tokens than Type 1, indicating a smaller TRF.

Acknowledgements

This work is partially supported by the National Key R&D Program of China (NO.2022ZD0160100).

References

Akbari, H.; Yuan, L.; Qian, R.; Chuang, W.-H.; Chang, S.-F.; Cui, Y.; and Gong, B. 2021. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. In *arXiv preprint arXiv:2104.11178*.

Beltagy, I.; Peters, M. E.; and Cohan, A. 2020. Longformer: The Long-Document Transformer. In *arXiv:2004.05150*.

Chi, T.-C.; Fan, T.-H.; Ramadge, P. J.; and Rudnicky, A. I. 2022. KERPLE: Kernelized Relative Positional Embedding for Length Extrapolation. *ArXiv*, abs/2205.09921.

Chi, T.-C.; Fan, T.-H.; and Rudnicky, A. I. 2022. Receptive Field Alignment Enables Transformer Length Extrapolation. *ArXiv*, abs/2212.10356.

Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734. Doha, Qatar: Association for Computational Linguistics.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *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)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Gao, L.; Biderman, S.; Black, S.; Golding, L.; Hoppe, T.; Foster, C.; Phang, J.; He, H.; Thite, A.; Nabeshima, N.; Presser, S.; and Leahy, C. 2020. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. In *arXiv preprint arXiv:2101.00027*.

Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; and Dauphin, Y. N. 2017. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, 1243–1252. PMLR.

Gong, Y.; Chung, Y.-A.; and Glass, J. 2021. AST: Audio Spectrogram Transformer. In *Proc. Interspeech 2021*, 571–575.

Gulati, A.; Chiu, C.-C.; Qin, J.; Yu, J.; Parmar, N.; Pang, R.; Wang, S.; Han, W.; Wu, Y.; Zhang, Y.; and Zhang, Z., eds. 2020. *Conformer: Convolution-augmented Transformer for Speech Recognition*.

Hao, D.; Mao, Y.; He, B.; Han, X.; Dai, Y.; and Zhong, Y. 2024. Improving Audio-Visual Segmentation with Bidirectional Generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Hershey, S.; Chaudhuri, S.; Ellis, D. P. W.; Gemmeke, J. F.; Jansen, A.; Moore, R. C.; Plakal, M.; Platt, D.; Saurous, R. A.; Seybold, B.; Slaney, M.; Weiss, R. J.; and Wilson, K. W. 2016. CNN architectures for large-scale audio classification. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 131–135.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. In *Conference on Empirical Methods in Natural Language Processing*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Liu, Z.; Li, D.; Lu, K.; Qin, Z.; Sun, W.; Xu, J.; and Zhong, Y. 2022. Neural architecture search on efficient transformers and beyond. *arXiv preprint arXiv:2207.13955*.

Liutkus, A.; Cífka, O.; Wu, S.-L.; Simsekli, U.; Yang, Y.-H.; and Richard, G. 2021. Relative positional encoding for transformers with linear complexity. In *International Conference on Machine Learning*, 7067–7079. PMLR.

Lu, K.; Liu, Z.; Wang, J.; Sun, W.; Qin, Z.; Li, D.; Shen, X.; Deng, H.; Han, X.; Dai, Y.; and Zhong, Y. 2022. Linear video transformer with feature fixation. *arXiv preprint arXiv:2210.08164*.

Merity, S.; Xiong, C.; Bradbury, J.; and Socher, R. 2016. Pointer Sentinel Mixture Models. In *arXiv:1609.07843*.

Ott, M.; Edunov, S.; Baevski, A.; Fan, A.; Gross, S.; Ng, N.; Grangier, D.; and Auli, M. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Press, O.; Smith, N.; and Lewis, M. 2022. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation. In *International Conference on Learning Representations*.

Qin, Z.; Han, X.; Sun, W.; He, B.; Li, D.; Li, D.; Dai, Y.; Kong, L.; and Zhong, Y. 2023a. Toeplitz Neural Network for Sequence Modeling. In *The Eleventh International Conference on Learning Representations*.

Qin, Z.; Han, X.; Sun, W.; Li, D.; Kong, L.; Barnes, N.; and Zhong, Y. 2022a. The Devil in Linear Transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 7025–7041. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Qin, Z.; Li, D.; Sun, W.; Sun, W.; Shen, X.; Han, X.; Wei, Y.; Lv, B.; Yuan, F.; Luo, X.; Qiao, Y.; and Zhong, Y. 2023b. Scaling TransNormer to 175 Billion Parameters. In *arXiv preprint 2307.14995*.

Qin, Z.; Sun, W.; Deng, H.; Li, D.; Wei, Y.; Lv, B.; Yan, J.; Kong, L.; and Zhong, Y. 2022b. cosFormer: Rethinking Softmax In Attention. In *International Conference on Learning Representations*.

Qin, Z.; Sun, W.; Lu, K.; Deng, H.; Li, D.; Han, X.; Dai, Y.; Kong, L.; and Zhong, Y. 2023c. Linearized Relative Positional Encoding. *arXiv preprint arXiv:2307.09270*.

Qin, Z.; Yang, S.; and Zhong, Y. 2023. Hierarchically gated recurrent neural network for sequence modeling. *NeurIPS*.

Qin, Z.; and Zhong, Y. 2023. Accelerating Toeplitz Neural Network with Constant-time Inference Complexity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Su, J.; Lu, Y.; Pan, S.; Wen, B.; and Liu, Y. 2021. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*.

Sun, J.; Zhong, G.; Zhou, D.; Li, B.; and Zhong, Y. 2022a. Locality Matters: A Locality-Biased Linear Attention for Automatic Speech Recognition. *arXiv preprint arXiv:2203.15609*.

Sun, W.; Qin, Z.; Deng, H.; Wang, J.; Zhang, Y.; Zhang, K.; Barnes, N.; Birchfield, S.; Kong, L.; and Zhong, Y. 2022b. Vicinity Vision Transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (01): 1–14.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wettig, A.; Gao, T.; Zhong, Z.; and Chen, D. 2022. Should You Mask 15% in Masked Language Modeling? In *arXiv:2202.08005*.

Zhu, Y.; Kiros, R.; Zemel, R.; Salakhutdinov, R.; Urtasun, R.; Torralba, A.; and Fidler, S. 2015. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In *The IEEE International Conference on Computer Vision (ICCV)*.