StableMask: Refining Causal Masking in Decoder-only Transformer

Qingyu Yin [1] **Xuzheng He** [2] **Xiang Zhuang** [1] **Yu Zhao** [3] **Jianhua Yao** [3] **Xiaoyu Shen** [4] **Qiang Zhang** [1]

Abstract

The decoder-only Transformer architecture with causal masking and relative position encoding (RPE) has become the *de facto* choice in language modeling. Despite its exceptional performance across various tasks, we have identified two limitations: First, it requires all attention scores to be non-zero and sum up to 1, even if the current embedding has sufficient self-contained information. This compels the model to assign disproportional excessive attention to specific tokens. Second, RPE-based Transformers are not universal approximators due to their limited capacity at encoding absolute positional information, which limits their application in position-critical tasks. In this work, we propose *StableMask*: a parameter-free method to address both limitations by refining the causal mask. It introduces pseudo-attention values to balance attention distributions and encodes absolute positional information via a progressively decreasing mask ratio. StableMask's effectiveness is validated both theoretically and empirically, showing significant enhancements in language models with parameter sizes ranging from 71M to 1.4B across diverse datasets and encoding methods. We further show that it naturally supports (1) efficient extrapolation without special tricks such as StreamingLLM and (2) easy integration with existing attention optimization techniques. [1]

1. Introduction

Large Language Models (LLMs) have revolutionized natural language processing for their task-agnostic in-context learning paradigm (Brown et al., 2020). The core of LLMs is the decoder-only Transformer architecture (Vaswani et al., 2017; Radford et al., 2019), characterized by the self-attention mechanism and relative positional encoding (RPE) to aggregate information and catch the dependency among tokens. It has exhibited superior zero-shot generalization capabilities in comparison to its encoder-decoder counterparts, leading to its increased prevalence in pre-trained LLMs (Lester et al., 2021; Patel et al., 2023). Despite the impressive success, we identified two important issues within this architecture.

The first issue arises from the softmax function used in self-attention, as its outputs consist solely of non-zero values summing up to 1 (Pang et al., 2019). This forces to allocate a certain distribution of attention probability across all available tokens, even when the current token already has sufficient self-contained information (Xiao et al., 2023) or when the attention mechanism does not need to prioritize any token (Hua et al., 2022; Bondarenko et al., 2023). In such cases, the model tends to allocate *disproportional attention* scores to specific tokens like punctuation marks. This problem is exacerbated in decoder-only models as the varied sequence length leads to an extremely uneven attention distribution, particularly on the initial tokens. While approaches have been proposed to mitigate this issue, they all entail significant complexity. e.g., modifying the sparseness of softmax (Laha et al., 2018), or adding dedicated tokens to absorb unnecessary attention (Darcet et al., 2023).

The second limitation is associated with various relative positional encoding strategies (Ke et al., 2020), e.g. ALiBi (Press et al., 2022), T5 (Raffel et al., 2020), and RoPE (Su et al., 2021). Compared with absolute position encoding (APE), RPE has achieved state-of-the-art performance in most natural language task. It also exhibits better extrapolation capabilities, and naturally preserves invariant properties for several important transformations like rotation and translation, making it more widely used in Transformers (Press et al., 2022). However, RPE fails to capture enough absolute positional information as the softmax always generates a right stochastic matrix (Luo et al., 2022), i.e., a square matrix where each row consists of non-negative real numbers adding up to 1. This restricts its application in situations where such positional information is crucial. Previous attempts to address this, such as URPE (Luo et al., 2022), added learnable relative position matrices atop the softmax outputs, which hurt the extrapolation capabilities

[1]Zhejiang University [2]Peking University [3]Tencent AI Lab [4]Eastern Institute of Technology, Ningbo. Correspondence to: Qiang Zhang <qiang.zhang.cs@zju.edu.cn>, Xiaoyu Shen <xyshen@eitech.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

[1]The code of this paper is available at https://github.com/MikaStars39/StableMask

because of the non-extensibility of learnable parameters.

In this paper, we propose *StableMask* – a tailored approach to address both issues by carefully modifying the causal mask in the decoder-based transformers. It introduces extra pseudo attention scores to the upper triangular attention matrix, which stabilizes the normalization constant of attention scores within each row regardless of the sequence length and token position. This allows the model to allocate excess attention to these dedicated pseudo scores. Moreover, StableMask progressively ensures that the result of softmax is not a right stochastic matrix. With a decreasing mask ratio (i.e. the sum of each row after softmax), it enables the model to encode a measurement of absolute position during the softmax stage, while remaining consistent with the decaying inter-token dependency used in RPE, thus effectively maintaining its extrapolation capability.

StableMask's effectiveness has been thoroughly validated through extensive testing on multiple language models across a diverse array of both synthetic and realistic tasks. It represents a substantial advancement in refining the attention mechanisms for decoder-only Transformers, overcoming the inherent limitations while retaining their core strengths. A key advantage of StableMask is its parameter-free nature. As StableMask is implemented solely as a direct replacement for the causal mask, it is highly compatibile with the Transformer's native architecture (such as different position encodings, attention optimizations or extrapolation techniques). For instance, we have presented an implementation of StableMask that is optimized for hardware efficiency, aligning with the principles of FlashAttention (Dao et al., 2022). This allows StableMask to seamlessly integrate into the ecosystem of Transformer models, thereby expanding its potential applications.

Our core contributions can be summarized as follows:

1. We identified two issues in the commonly used decoder-only Transformer architecture: the disproportional attention distribution and the inability to accurately capture positional information.
2. We propose StableMask, an efficient and easily integrable solution to effectively address both issues by carefully modifying the causal mask.
3. We validate the effectiveness of StableMask across multiple tasks and encoding methods.
4. We present a hardware-efficient version of StableMask to optimize its practical applicability.

2. Preliminary

Self-Attention Let X be the input sequence, n be the sequence length and d be the dimensionality of the hidden state. The self-attention mechanism in Transformer architectures calculates attention scores between each pair of words to capture dependencies between words and learn contextual information effectively. Let A denote the attention score matrix and a_{ij} be the attention score between the i-th word and the j-th word. We have $A = \frac{QK^\top}{\sqrt{d}}$ where $Q, K, V \in \mathbb{R}^{n\times d}$ represent the Query, Key, and Value matrices derived from X (Vaswani et al., 2017). In decoder-only models, A is further modified by a causal mask M and a softmax operation $\tilde{A} = \mathrm{Softmax}(A + M)$. The following holds to prevent the model from attending to future tokens:

$$M_i = [\underbrace{0, \cdots, 0}_{i}, \underbrace{-\infty, \cdots, -\infty}_{n-i}]_n, \quad (1)$$

$$\tilde{A}_i = [a_{i1}, a_{i2}, \cdots, a_{ii}, 0, \cdots, 0]_n. \quad (2)$$

Position Encoding The raw Transformer without position encodings is insensitive to permutational rearrangements. Two chief methods have been employed to remove this insensitivity: absolute position encoding (APE) and relative position encoding (RPE). APE assigns an index-dependent vector at each position to the word embeddings. These assigned vectors are usually trainable parameters to represent absolute positions of each input token (Kenton & Toutanova, 2019; Radford et al., 2019). More recently, RPE such as ALiBi (Press et al., 2022), T5 (Raffel et al., 2020) and RoPE (Su et al., 2021) took a different approach by incorporating relative distances of positions into the attention score matrix. RPEs can be mainly classified into additive (T5, ALiBi, etc.) or multiplicative (RoPE, etc.):

$$\text{Add:} \quad \tilde{A}_{\text{add}} = \mathrm{Softmax}\left(\frac{QK^\top + S}{\sqrt{d_k}} + M\right), \quad (3)$$

$$\text{Mul:} \quad \tilde{A}_{\text{mul}} = \mathrm{Softmax}\left(\frac{\tilde{Q}\tilde{K}^\top}{\sqrt{d_k}} + M\right), \quad (4)$$

where $\tilde{Q} = Q \odot R_Q,\ \tilde{K} = K \odot R_K$. R_Q, R_K are rotary forms usually in complex values and S is a Topelitz matrix. Given its consistent demonstrated improvements over APE, RPE has emerged as the default choice in LLMs.

3. Problem

Despite the exceptional performance, we identified two key issues associated with self-attention and RPE.

Disproportional Attention The first issue arises from the softmax function used in self-attention. Given that the softmax function requires all attention scores to be non-zero and sum up to 1, it necessitates an inescapable distribution of attention across on all visible tokens. However, previous studies (Shen et al., 2019; Hassid et al., 2022; Bondarenko et al., 2023; Xiao et al., 2023; Hu et al., 2024) have shown

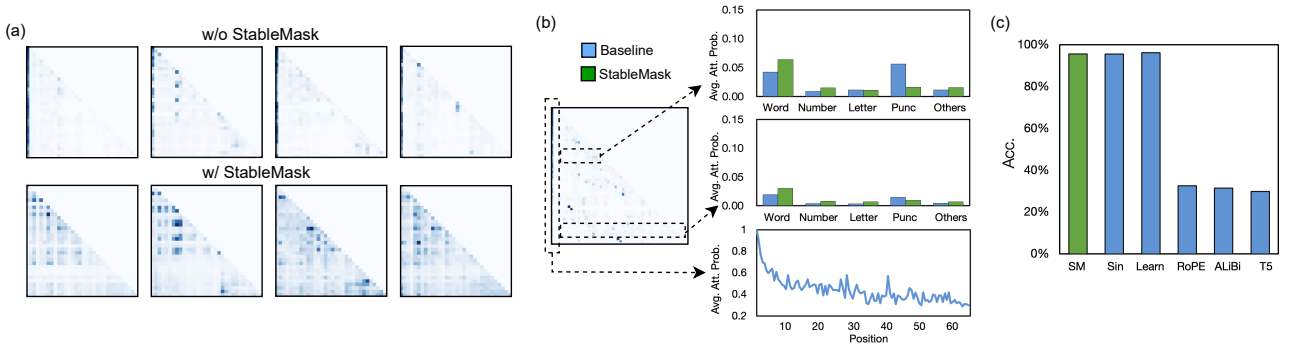

Figure 1. (a) Visual comparison of attention heads with and without StableMask on the OpenLLaMA 1.4B model. (b) The attention allocation to various types of tokens (excluding the initial token) at two different positions and the trend of attention allocation to the initial token over positions, *averaged over heads*. Blue: The original Transformer exhibits a clear disproportional attention issue. Green: StableMask effectively rectifies the proportion of attention allocation. (c) Experimental Results showing RPE's inability to encode absolute position (Blue). StableMask solves the issue of RPE's inability to encode absolute position (Green).

that the attention mechanism often requires very few important tokens, and the others are merely distractions. In this case, the requirement imposed by the softmax function prevents the model from effectively zeroing out the attention scores for irrelevant tokens. Some of these irrelevant tokens, such as initial tokens or non-functional words like punctuation marks, are more frequently observed by other tokens. In consequence, as shown in Figure 1, the model tends to allocate disproportional attention (DA) to them. We refer to these tokens which are not semantically relevant, but receive disproportional attention values, as DA tokens[2]. The existence of DA tokens can lead to various undesired problems, e.g., perplexity surge in length extrapolation or sensitivity to irrelevant noise (Xiao et al., 2023).

Interestingly, the extent of this DA phenomenon varies across token positions within the decoder-only language model. It is most prominent at the beginning of a sequence, and gradually eases towards the end (as seen in Figure 1(b)). Intuitively, as the token position increases, more tokens participate in the softmax operation and even assigning a very small probability to each token can result in a significant accumulative probability. As a result, DA tokens cannot receive as much attention values as they do near the beginning of a sequence.

Existing solutions, such as StreamingLLM (Xiao et al., 2023) and ViT Register (Darcet et al., 2023), have attempted to address this by introducing *Artificial Tokens* (AT) to absorb excess attention, so that real tokens can be freed from getting unnecessary DA. We term them as AT-based methods. However, as said, the severity of the DA issue varies along token positions. We hypothesize that adding a fixed number of tokens across all sequences is not position-adaptive and thereby cannot fully address the DA issue.

Inability to Encode Absolute Position Despite its superior performance, RPE that modifies $QK^\top$ does not ensure V is also sensitive to position. For instance, when all inputs are identical vectors, the outputs are also guaranteed to be equal because the output of softmax generates a right stochastic matrix [3]. Therefore, RPE can perform poorly in tasks where positional information is critical.

To verify this limitation of RPEs, we designed specialized datasets, inspired by URPE (Luo et al., 2022), which focus on tasks requiring absolute positional information while maintaining consistent input sequences (check Appendix B.2 for details). We report the average accuracy of various models in Figure 1(c). The results demonstrate that models relying exclusively on RPEs exhibit poor performance, confirming the inferiority of RPE in capturing absolute positional information.

One obvious solution to the limitation is to directly replace RPE with APE. However, as mentioned, APE has its own problems such as poor extrapolation, rotation and translation variant, worse prediction accuracy, etc (Su et al., 2021; Press et al., 2022). Another approach is to add additional parameters to the matrix after the softmax to re-encode absolute positional information. For example, URPE (Luo et al., 2022) adds a learnable Toeplitz matrix $\mathcal{T}$ to the softmax matrix $\tilde{A}$ via:

$$\text{Attention}(Q, K, V) = (\tilde{A} \odot \mathcal{T})V. \tag{5}$$

The URPE approach, while successfully encoding absolute positional information, has several drawbacks. First, it requires additional learnable parameters which complicates

[2]Appendix A offers an information-theoretic definition and interpretation of the DA issue.

[3]For a more in-depth discussion on all-identical inputs and their relation to DA, refer to Appendix B.1.

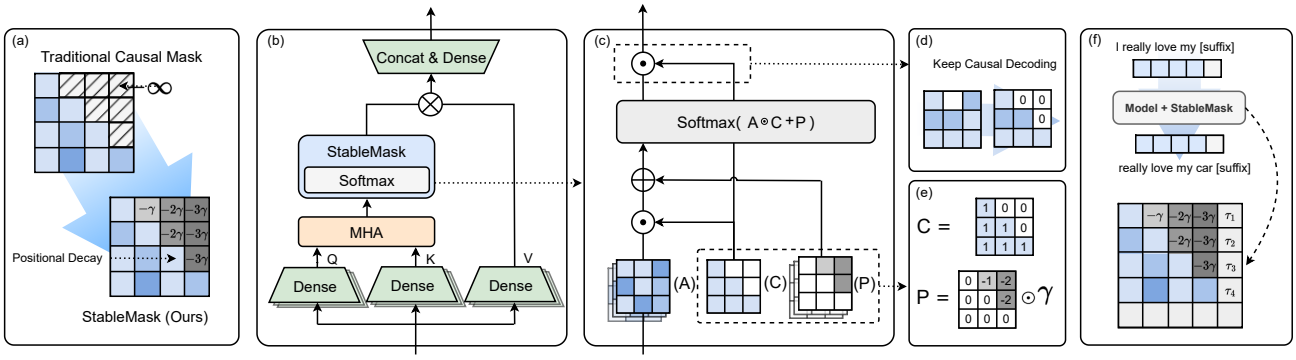

Figure 2. (a) Illustration of the StableMask mechanism. (b) StableMask integrates with the softmax operation, replacing the traditional causal mask. (c) The attention score matrix is first cleared of attention values in the upper triangular part using the C matrix, then pseudo-attention scores are added using the P matrix followed by the softmax computation. (d) After the softmax operation, the remaining attention probabilities in the upper triangular part are cleared using C to ensure the causal decoding property. (e) The C matrix has zeros in the upper triangular part and ones in the lower triangular part, while the P matrix has linear decay in the upper triangular part and zeros in the lower triangular part. γ is a hyperparameter. (f) StableMask for inference. An input sequence needs a suffix.

the model optimization. Second, because the $\mathcal{T}$ matrix is fixed, models trained with this method loses its ability to input context that is longer than the training length.

4. StableMask

In the previous section, we analyzed two problems with the decoder-only Transformer architecture commonly used in contemporary LLMs: disproportional attention and inability to encode absolute position. Disproportional attention happens when certain attention heads share no need to allocate any attention logits but have to due to the softmax mechanism, and this issue is more pronounced at the beginning of the sequence in the decoder. The inability to encode absolute position comes from the result of softmax: it is a right stochastic matrix, with the sum of each row equals one always, so its output is insensitive to absolute positions.

To address the above two problems, we seek a solution by introducing pseudo-attention scores into the softmax operation. Specifically, the solution should simultaneously meet the following requirements:

(i) It can provide *additional pseudo-attention scores* to accommodate excess attention logits, thereby freeing DA tokens from the responsibility of absorbing unnecessary attention values.

(ii) These additional pseudo-attention scores need to adhere to the property of DA in a decoder-only model, i.e. larger at the beginning of the sequence and smaller towards the end of the sequence.

(iii) It ensures that the result of softmax is not a right stochastic matrix, i.e. the sum of each row is not 1, so that positional information can be encoded.

In the following section, we show that all of the above three requirements can be met by carefully modifying the causal mask applied after softmax.

4.1. Pseudo-attention Score

To meet the requirement (i) and (ii), we propose constructing a StableMask attention score matrix $A_{\text{SM}} \in \mathbb{R}^{n\times n}$:

$$A_{\text{SM}} = \begin{pmatrix} a_{11} & p_{11} & \cdots & p_{1(n-1)} \\ a_{21} & a_{22} & \cdots & p_{1(n-2)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}. \tag{6}$$

Here, we call these p_{ij} as *pseudo-attention scores*. When the current attention head does not depend too much on its previous context, it can choose to store unnecessary attention values on these pseudo-attention scores. For each row (all attention scores for the i-th token), the sequence length it can attend to is fixed to be n. Therefore there will be $n-i$ pseudo-attention scores in each row for excessive attention allocation. This fulfills requirement (ii), which involves having more pseudo-attention values towards the beginning of a sequence. A_{SM} can be calculated using the following method:

$$A_{\text{SM}} = A \odot C + P, \tag{7}$$

$$C = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}, P = \begin{pmatrix} 0 & p_{11} & \cdots & p_{1(n-1)} \\ 0 & 0 & \cdots & p_{1(n-2)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}.$$

The problem then becomes how should the values of these pseudo-attention scores be set. At the start of training, the

distribution of the scaled attention scores has a mean of 0. These attention scores are also influenced by position encoding, and commonly used RPEs typically exhibit decay with increasing relative distance. Therefore, pseudo-attention scores should not significantly disrupt the original distribution of attention scores, and they should also align with the characteristics of the relative position encoding used by the model. Consequently, for p_{ij}, it should conform to:

$$p_{\text{base}} = 0, \;\; p_{ij} = p_{\text{base}} - (j-1)\gamma, \tag{8}$$

where γ is a decay rate hyperparameter. Therefore, the attention score matrix with StableMask should be:

$$A_{\text{SM}} = \begin{pmatrix} a_{11} & -\gamma & \cdots & -(n-1)\gamma \\ a_{21} & a_{22} & \cdots & -(n-1)\gamma \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}. \tag{9}$$

Finally, we can replace the traditional causal mask operation in Equation (2) with:

$$\begin{aligned} \tilde{A} &= \text{Softmax}(A_{\text{SM}}) \odot C \\ &= \text{Softmax}(A \odot C + P) \odot C. \end{aligned} \tag{10}$$

Here the $A_{\text{SM}} = A \odot C + P$ inside Softmax masks the attention score matrix with pseudo-attention scores, whereas the C outside Softmax replaces the scores which need masking with 0 again. Therefore, StableMask still maintains the characteristics of causal decoding, ensuring that information does not leak from subsequent tokens.

4.2. StableMask Encodes Absolute Position

StableMask introduces a set of pseudo-attention scores. Therefore, for those real attention scores (the lower triangular part of the attention matrix A_{SM}), their sum after softmax will not be 1, meeting the requirement (iii). Concretely, let A_i denote the real attention scores of the i-th row and P_i denote the pseudo-attention scores in the i-th row, we have:

$$\sum \text{Softmax}_{A_i \bigcup P_i}(A_i) = 1 - \sum \text{Softmax}_{A_i \bigcup P_i}(P_i),$$

where $\text{Softmax}_{A_i \bigcup P_i}(A_i)$ and $\text{Softmax}_{A_i \bigcup P_i}(P_i)$ are the real/pseudo attention in each row. We reconsider the question posed in Section 3: whether the model can encode positional information for an identical input sequence $X = [\boldsymbol{x}, \cdots, \boldsymbol{x}]_n$. The answer is affirmative: notice that $\Sigma_{j \le i} \exp(A_{ij})$ increases as i increases (all A_{ij}s are equal), and $\Sigma_{j>i} \exp(P_{ij})$ decreases as i increases, we have

$$\sum \text{Softmax}_{A_i \bigcup P_i}(A_i) < \sum \text{Softmax}_{A_{i+1} \bigcup P_{i+1}}(A_{i+1}),$$

which means after Equation (10), the output attention values will be monotonic:

$$\begin{aligned} \tilde{A}(W_V X)^\top &= [\alpha_1 \boldsymbol{v}, \alpha_2 \boldsymbol{v}, \cdots, \alpha_n \boldsymbol{v}]_n, \\ 0 < \alpha_1 &< \alpha_2 < \cdots < \alpha_n = 1. \end{aligned}$$

This indicates that absolute positional information is effectively captured.

In general, a Transformer decoder with StableMask has the ability to encode absolute positional information:

Theorem 4.1. *Let $X = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n]_n$ be an input sequence of length n to the StableMask model $f_T^{(SM)}$. Then, the first layer of $f_T^{(SM)}$ can recover absolute positions $[1, 2, \ldots, n]$ in the hidden state $\Omega^{(1)}$. That is, there exist W_Q, W_K, W_V and W_O for the first attention layer, along with W_1 and W_2 for the first feed-forward layer, that computes absolute positions and pass them to the next layer.*

The complete proof can be found in Appendix C.

4.3. Inference and Length Extrapolation

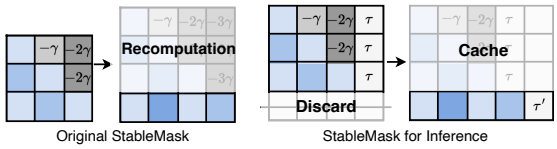

Figure 3. StableMask for Inference. The original StableMask implementation needs to recompute the softmax result for the attention score matrix because additional mask values are added. StableMask for Inference introduces a factor τ to fix the situation to be in the form of the maximum training length.

In Section 4.1, we introduced the computation process of StableMask. During the training phase, StableMask can be readily applied in parallel within a batch to backpropagate the training loss. During inference, attention computation is usually performed serially and employs KV caching (Tang et al., 2021; Pope et al., 2023).

StableMask in its original form is not cost-effective for inference, because it does not support the use of KV caching. During the inference stage, when the sequence length is changed e.g. from n to $n+1$ for causal decoding, attention layers need to recalculate the softmax results. For the first n rows, an additional pseudo-attention value is added, invalidating the previously calculated attention (see Figure 3). This renders KV caching unusable, significantly increasing the cost of inference. Our solution is simple: we pad the sequence to the training length while compressing the padded tokens into a single suffix token. Assuming the current sequence length is n, we first append a suffix token to the end of the sequence (See Figure 2 (f) and Figure 3). At this point, the size of the attention matrix becomes $(n+1) \times (n+1)$. Then, in the additional last column, we

WikiText-103				MiniPile				
Model	*PE	#Params	PPL	Model	*PE	#Params	PPL 1 Epoch	PPL 2 Epoch
BLOOM	ALiBi	71M	$29.9_{\pm.1}$	BLOOM	ALiBi	160M	$25.8_{\pm.2}$	$23.3_{\pm.4}$
BLOOM-SM	ALiBi	71M	$\mathbf{29.0}_{\pm.1}$	BLOOM-SM	ALiBi	160M	$\mathbf{25.6}_{\pm.0}$	$\mathbf{22.9}_{\pm.2}$
OpenLLaMA	RoPE	71M	$27.4_{\pm.2}$	OpenLLaMA	RoPE	160M	$25.9_{\pm.1}$	$21.2_{\pm.1}$
OpenLLaMA-SM	RoPE	71M	$\mathbf{26.9}_{\pm.3}$	OpenLLaMA-SM	RoPE	160M	$\mathbf{25.0}_{\pm.0}$	$\mathbf{20.9}_{\pm.3}$
BLOOM	ALiBi	160M	$27.6_{\pm.9}$	BLOOM	ALiBi	430M	$20.6_{\pm.1}$	$15.6_{\pm.4}$
BLOOM-SM	ALiBi	160M	$\mathbf{26.1}_{\pm.2}$	BLOOM-SM	ALiBi	430M	$\mathbf{19.6}_{\pm.3}$	$\mathbf{15.5}_{\pm.2}$
OpenLLaMA	RoPE	160M	$22.5_{\pm.8}$	OpenLLaMA	RoPE	430M	$19.6_{\pm.2}$	$15.7_{\pm.5}$
OpenLLaMA-SM	RoPE	160M	$\mathbf{21.1}_{\pm.6}$	OpenLLaMA-SM	RoPE	430M	$\mathbf{19.5}_{\pm.4}$	$\mathbf{15.1}_{\pm.5}$

*: *positional encoding type*

Table 1. Pretraining results with (" -SM") or without StableMask on the Wikitext-103 and MiniPile datasets.

add a factor $\tau = \ln(\sum_{i=n}^{N-1} e^{-i\gamma})$:

$$A'_{\text{SM}} = \begin{pmatrix} a_{11} & -\gamma & \cdots & -(n-1)\gamma & \tau \\ a_{21} & a_{22} & \cdots & -(n-1)\gamma & \tau \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & \tau \\ a_{(n+1)1} & a_{(n+1)2} & \cdots & a_{(n+1)n} & a_{(n+1)(n+1)} \end{pmatrix}.$$

The last row of A'_{SM} comes from the suffix and will not be utilized for generation. This makes each row equivalent to the case when the sequence length is the same as the training length, allowing us to use KV caching.

Next, we deal with the length extrapolation scenario, i.e. inputs that are longer than the pretraining length limit. Notice that when n reaches the maximum training length N, τ becomes 0. This setup prevents the model from continuing to generate τ values beyond the training length. Therefore, during extrapolation, we set $\tau = -n\gamma$, where $n \geq N$ is the current sequence length. τ in long sequences is a very small number after applying the softmax, and its value will approach zero as n grows. However, the presence of this term still ensures that the softmax result is not a right stochastic matrix, thereby asymptotically encoding absolute positional information. In addition, when the sequence length is very long, the phenomenon of disproportional attention nearly disappears, as we concluded in Section 3. Hence the pseudo-attention score does not need to maintain a large value.

4.4. Hardware-Efficient Implementation of StableMask

FlashAttention (Dao et al., 2022) represents a major advance in accelerating the Transformer architecture. It avoids repeated data transfers between GPU's High Bandwidth Memory (HBM) and processing units, by segmenting and sequentially processing the QKV matrix on-chip. StableMask's integration into this framework is seamless, requiring only minimal modifications. In the FlashAttention paradigm, the query $Q \in \mathbb{R}^{n \times d_H}$, key $K \in \mathbb{R}^{n \times d_H}$, and value $V \in \mathbb{R}^{n \times d_H}$ matrices are partitioned into $Tr = \frac{n}{Br}$ blocks $Q_1, \ldots, Q_{Tr}$, $K_1, \ldots, K_{Tr}$, $V_1, \ldots, V_{Tr}$, each of dimension $\mathbb{R}^{Br \times d_H}$. Then each block Q_i, K_j, V_i is fetched for computation. The attention scores $S_i^{(j)}$ for blocks Q_i and K_j are derived from the on-chip computation: $S_i^{(j)} = Q_i K_j^T \in \mathbb{R}^{Br \times Br}$. With the incorporation of StableMask into FlashAttention, two additional fused operations are introduced as follows:

$$S_i^{(j)} = (Q_i K_j^T) \odot C_i^{(j)} + P_i^{(j)}, \tag{11}$$

where P and C correspond to the StableMask matrices, segmented into $Tr \times Tr$ blocks with $P_i^{(j)}, C_i^{(j)} \in \mathbb{R}^{Br \times Br}$ and loaded on-chip. We include a complete formula derivation and pseudocode implementation in Appendix D.

5. Experiments

In this section, we present extensive experiments to rigorously evaluate the performance of our proposed method.

5.1. StableMask Solves Two Problems

Our initial assessment confirms the efficacy of the StableMask model in addressing the two problems in Transformer models. The experimental results have been presented in Figure 1. Firstly, concerning the disproportionate attention problem, we perform a comparative visualization of the attention heads in models with and without StableMask. By calculating the attention probability ratios for the first token and various token types, we observed that StableMask largely rectifies the issue of abnormal attention distribution. With StableMark, both initial tokens and punctuation marks experience a significant reduction in attention values. Regarding the second issue of encoding absolute positional information, we evaluated the model's fitting capabilities on a specially designed dataset, comparing StableMask with various Position Encoding approaches. The findings indicate StableMask adeptly encodes absolute positional information, thereby effectively remedying the limitations inherent in relative position encoding. We also provided a visualiza-

Model	PPL / Tokens					DownStream Tasks					
	5B	10B	15B	20B	25B	LBD	PIQA	ARCE	ARCC	OBQA	WG
OpenLLaMA	$15.4_{\pm.2}$	$14.8_{\pm.3}$	$12.4_{\pm.3}$	$11.7_{\pm.2}$	$10.7_{\pm.3}$	59.4	67.1	51.4	25.6	31.4	53.5
OpenLLaMa-SM	$\mathbf{15.0}_{\pm.2}$	$\mathbf{14.6}_{\pm.1}$	$\mathbf{11.9}_{\pm.1}$	$\mathbf{11.3}_{\pm.4}$	$\mathbf{10.4}_{\pm.3}$	**59.6**	67.1	**51.7**	25.6	**32.6**	**54.1**

Table 2. Left: Pretraining result of OpenLLaMA 1.4B with RoPE. Right: Result of downstream tasks on OpenLLaMA 1.4B.

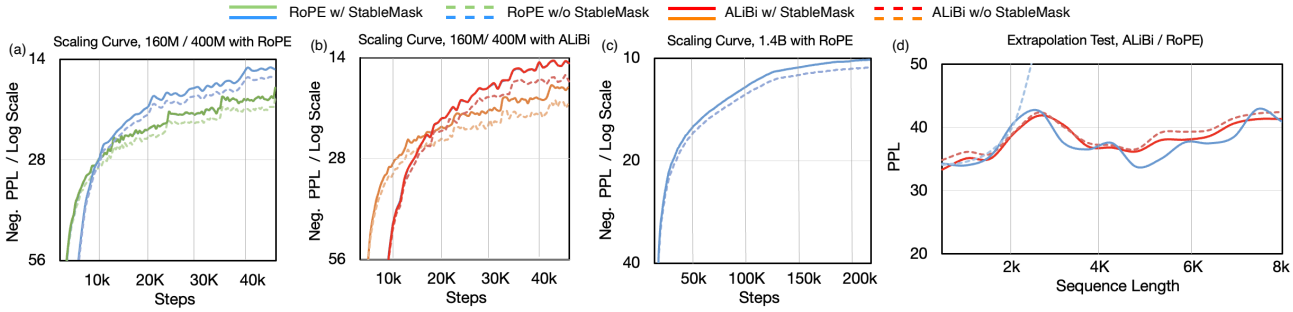

Figure 4. (abc): Scaling Curve of models from 160M to 1.4B across different positional encodings. (d): extrapolation results (with window attention). StableMask consistently improves the model performance while enabling effective extrapolation.

tion of the new attention score matrix after softmax with StableMask in Appendix G.

5.2. StableMask Improves Model Performance

We further tested the performance of StableMask on various model architectures and position encodings. Our experiments leverage models built on BLOOM (LLaMA architecture with ALiBi) and OpenLLaMA (Touvron et al., 2023) (RoPE (Su et al., 2021)) architectures. Detail settings could be checked in the Appendix F.

Performance on Wikitext-103 and MiniPile (Table 1): Empirical evidence underscores the efficacy of models employing StableMask when trained on both Wikitext-103 (Merity et al., 2016) and MiniPile (Kaddour, 2023). These models demonstrate enhanced perplexity (PPL) scores, a pattern consistent across different architectures and sizes, including those with ALiBi and RoPE, and spanning parameter scales of 71M to 400M. Notably, within those datasets, models integrating StableMask consistently outshine their counterparts lacking this feature.

Impact on Scaling Performance (Table 2): The Pile is an extensive open-source dataset tailored for large-scale language modeling. We pretrained a 1.4B model with LLaMA architecture on the Pile dataset with 25B tokens. In the context of scaling of tokens, the model with StableMask consistently achieves better PPL scores compared to the standard OpenLLaMA model, showing the scaling ability of models with StableMask.

Effectiveness in Downstream Tasks (Table 2): When examining pre-trained models on downstream tasks like LAMBADA (Paperno et al., 2016), PIQA (Bisk et al., 2019), ARC-Easy (Yadav et al., 2019), ARC-Challenge (Yadav et al., 2019), OpenbookQA (Mihaylov et al., 2018), and Winogrande (Sakaguchi et al., 2021), model with StableMask shows a general trend of improved performance. It suggests that StableMask not only improves language understanding in the pretraining stage but also enhances effectiveness in downstream tasks.

5.3. Extrapolation Capability

As StableMask resolves the problem of DA tokens, it naturally addresses the attention sink issue (Xiao et al., 2023), where initial tokens get large attention values and removing them from the attention window leads to a surge in perplexity. The models with our proposed StableMask do not need to preserve tokens at the beginning of the sequence during window-based extrapolation and avoid causing generation failures. As shown in Figure 4, when using the RoPE position encoding, the extrapolation perplexity quickly explodes without StableMask. When StableMask is applied, the extrapolation perplexity remains stable with window attention, where only the most recent KVs are cached. Furthermore, we believe that the parameter-free nature of StableMask facilitates its seamless integration with other extrapolation methods, a prospect we leave for future exploration.

5.4. StableMask vs AT-based Methods

In Section 3, we discussed that the artificial token (AT)-based methods are one alternative method to mitigate the DA problem. These artificial tokens could be either learnable, i.e. added before the embedding layer, or fixed as

Methods	PPL	Pseudo Value	PPL
Baseline	22.5	$-\infty$	22.5
Learnable AT	21.6	0	21.5
Fixed Value AT	22.4	1×10^{-2}	22.2
StableMask	**21.1**	Positional Decay	**21.1**

Table 3. Left: Experiment result of ablation study and comparison of AT method on OpenLLaMA, 160M. Right: Ablation experiment, 160M on OpenLLaMA.

constant vectors, e.g. zero vector. However, we find that as AT-based methods provide the same number of tokens for all sequences, its benefit is not as significant as StableMask (see Table 3) since the severity of the DA issue varies along the sequence. For fair comparison, we retrained OpenLLaMA models using the AT method and StableMask on the MiniPile dataset.

5.5. Impact on Inference Efficiency

In Section 4.3, we introduced StableMask for Inference, which changes the form of the mask to allow for more efficient inference strategies like KV cache. To validate its effectiveness, we tested the inference efficiency of a standard Transformer (Baseline), a model using StableMask (SM), and a model using StableMask for Inference (SM-I). We present the results in Figure 5 and find that StableMask for Inference significantly improved the model's inference efficiency, making it comparable to the efficiency of traditional Transformers.

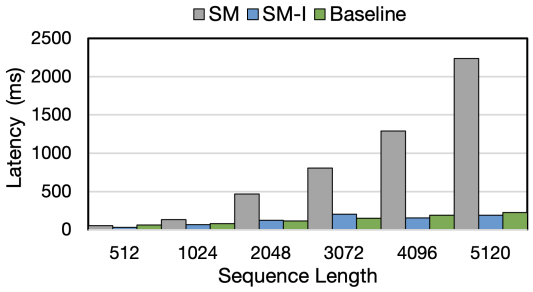

Figure 5. Inference latency test on OpenLLaMA 1.4B. Our proposed StableMask adapted for fast inference (SM-I) significantly reduces the running latency.

5.6. Effects of Pseudo Attention Value

In Section 4, we introduced positional linear decay, making the pseudo-attention scores align with the characteristics of real attention scores.w To validate its rationality, we conducted ablation experiments on various types of pseudo-attention scores. These experiments included four modes: (a) No addition of pseudo-attention scores, i.e., maintaining a mask of negative infinity. (b) Padding with zeros, which aligns with the values of attention score distribution. (c) Padding with a value different from the attention score distribution, e.g. 1×10^{-2}. (d) The positional decay method we proposed.

Our ablation studies, as detailed in Table 3, demonstrate that a decay value like 1×10^{-2} deviates significantly from the original attention matrix's distribution, leading to diminished pretraining performance. The implementation of positional decay, however, excels in the training phase, showcasing state-of-the-art performance.

6. Related Work

Several studies have attempted to address issues inherent in the attention mechanism and softmax operation. A pivotal contribution by (Hassid et al., 2022) raised questions about the role of certain heads in the attention mechanism. They discovered that substituting a subset of heads with constant diagonal matrices could even enhance model performance, suggesting that part of the model's attention heads do not need to attend to any tokens other than themselves. Quantizable Transformer (Bondarenko et al., 2023) and StreamingLLM (Xiao et al., 2023) identified a tendency in some attention heads to accumulate probabilities on the initial few tokens or on tokens similar to punctuation marks. Bondarenko et al. (2023) demonstrated that this behavior impacts model quantization, proposing a solution by trimming softmax and employing gated attention. StreamingLLM, on the other hand, observed that this phenomenon affects windowed attention, and addressed it by preserving the initial tokens. Darcet et al. (2023) proposed adding "register tokens" which are essentially artificial places for the real tokens to attend to. The added tokens serve as a way to absorb the excessive attention that would otherwise accumulate on the initial tokens.

However, the previous approach of adding or using extra tokens either (1) uses fixed values or weights which does not account for possible distributional shifts when extrapolating to longer sequences; (2) does not explore its potential interference with positional embeddings; (3) adds extra parameters or computation to the attention layer, while not making clear whether existing optimization techniques are still applicable; (4) does not provide a theoretical framework for understanding the phenomenon more deeply.

7. Conclusion

StableMask represents a significant advancement in the field of language modeling, by simultaneously addressing two limitations of the decoder-only Transformer architecture: disproportional attention and inability to encode absolute position. By refining the causal mask with pseudo-attention values, StableMask adeptly balances attention distributions and encodes absolute positional information through a pro-

gressively decreasing mask ratio. It preserves the inherent distribution of the attention score matrix and enhances the model's ability in various natural language tasks.

While StableMask demonstrates much potential, it is not without its constraints. One notable limitation is the slightly increased computational demand compared to conventional attention mechanisms. However, as the increased computation is only one matrix multiplication, we believe this overhead is negligible. Furthermore, StableMask inherently encodes absolute positional information, necessitating careful calibration to prevent the model from being adversely affected. We anticipate that forthcoming research will further refine our approach and overcome these challenges.

Acknowledgement

We thank Songlin Yang and other collaborators for the suggestions on language expression and image design in this paper. This work is supported by New Generation AI Development Plan for 2030 of China (2023ZD0120802), National Natural Science Foundation of China (62302433, U23A20496), Zhejiang Provincial "Jianbing" "Lingyan" Research and Development Program of China (2024C01135), Zhejiang Provincial Natural Science Foundation of China (LQ24F020007) and CCF-Tencent Rhino-Bird Fund (RAGR20230122).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Bisk, Y., Zellers, R., Bras, R. L., Gao, J., and Choi, Y. Piqa: Reasoning about physical commonsense in natural language, 2019.

Bondarenko, Y., Nagel, M., and Blankevoort, T. Quantizable transformers: Removing outliers by helping attention heads do nothing. *arXiv preprint arXiv:2306.12929*, 2023.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33: 1877–1901, 2020.

Dao, T., Fu, D., Ermon, S., Rudra, A., and Ré, C. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

Darcet, T., Oquab, M., Mairal, J., and Bojanowski, P. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023.

Hassid, M., Peng, H., Rotem, D., Kasai, J., Montero, I., Smith, N. A., and Schwartz, R. How much does attention actually attend? questioning the importance of attention in pretrained transformers, 2022.

Hu, J. Y.-C., Chang, P.-H., Luo, R., Chen, H.-Y., Li, W., Wang, W.-P., and Liu, H. Outlier-efficient hopfield layers for large transformer-based models. *arXiv preprint arXiv:2404.03828*, 2024.

Hua, W., Dai, Z., Liu, H., and Le, Q. Transformer quality in linear time. In *International Conference on Machine Learning*, pp. 9099–9117. PMLR, 2022.

Kaddour, J. The minipile challenge for data-efficient language models, 2023.

Kazemnejad, A., Padhi, I., Ramamurthy, K. N., Das, P., and Reddy, S. The impact of positional encoding on length generalization in transformers, 2023.

Ke, G., He, D., and Liu, T.-Y. Rethinking positional encoding in language pre-training. *arXiv preprint arXiv:2006.15595*, 2020.

Kenton, J. D. M.-W. C. and Toutanova, L. K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019.

Kim, J., Kim, M., and Mozafari, B. Provable memorization capacity of transformers. In *International Conference on Learning Representations*, 2023.

Laha, A., Chemmengath, S. A., Agrawal, P., Khapra, M., Sankaranarayanan, K., and Ramaswamy, H. G. On controllable sparse alternatives to softmax. *Advances in Neural Information Processing Systems*, 31, 2018.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021.

Luo, S., Li, S., Zheng, S., Liu, T.-Y., Wang, L., and He, D. Your transformer may not be as powerful as you expect. *Advances in Neural Information Processing Systems*, 35: 4301–4315, 2022.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Conference on Empirical Methods in Natural Language Processing*, 2018. URL https://api.semanticscholar.org/CorpusID:52183757.

Pang, T., Xu, K., Dong, Y., Du, C., Chen, N., and Zhu, J. Rethinking softmax cross-entropy loss for adversarial robustness. In *International Conference on Learning Representations*, 2019.

Paperno, D., Kruszewski, G., Lazaridou, A., Pham, Q. N., Bernardi, R., Pezzelle, S., Baroni, M., Boleda, G., and Fernández, R. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*, 2016.

Park, S., Yun, C., Lee, J., and Shin, J. Minimum width for universal approximation. In *International Conference on Learning Representations*, 2021.

Patel, A., Li, B., Rasooli, M. S., Constant, N., Raffel, C., and Callison-Burch, C. Bidirectional language models are also few-shot learners. In *The Eleventh International Conference on Learning Representations*, 2023.

Polyanskiy, Y. and Wu, Y. Strong data-processing inequalities for channels and bayesian networks, 2016.

Pope, R., Douglas, S., Chowdhery, A., Devlin, J., Bradbury, J., Heek, J., Xiao, K., Agrawal, S., and Dean, J. Efficiently scaling transformer inference. *Proceedings of Machine Learning and Systems*, 5, 2023.

Press, O., Smith, N., and Lewis, M. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Shen, X., Zhao, Y., Su, H., and Klakow, D. Improving latent alignment in text summarization by generalizing the pointer generator. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 3762–3773, 2019.

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

Tang, Z., Li, C., Ge, J., Shen, X., Zhu, Z., and Luo, B. Ast-transformer: Encoding abstract syntax trees efficiently for code summarization. In *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 1193–1195. IEEE, 2021.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Xiao, G., Tian, Y., Chen, B., Han, S., and Lewis, M. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

Yadav, V., Bethard, S., and Surdeanu, M. Quick and (not so) dirty: Unsupervised selection of justification sentences for multi-hop question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 2019. doi: 10.18653/v1/d19-1260. URL http://dx.doi.org/10.18653/v1/D19-1260.

Yun, C., Bhojanapalli, S., Rawat, A. S., Reddi, S. J., and Kumar, S. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2020.

A. Detailed Explanation of the DA Issue

The traditional dot-product attention makes the assumption that the next token is strongly related to the previous context. However, the mutual information $I(X_{\leq i}; X_{n+1}) = H(X_{n+1}) - H(X_{n+1}|X_{\leq i})$ could be small, especially in the initial parts of the sequence. We formalize this (counter-)intuition by defining the following concepts:

Definition A.1. A *causally isotropic* data distribution of N discrete random variables $X_1, X_2, \ldots, X_N$ satisfies that for any set of indices $\Lambda \subset [n]$, $H(X_{n+1}|X_\Lambda = x_\Lambda) = H(X_{n+1}|X_\Lambda)$ does not depend on the value of x_Λ, where H denotes entropy[4].

Definition A.2. A *layer-wise* decoder for a data distribution $p(X_1, X_2, \ldots, X_N)$ accepts any data point $x_{<N}$, and computes deterministically L layers of intermediate representations $\Omega^{(l)}_{<N}$, such that for $n < N$, $\Omega^{(l)}_n$ only receives inputs from $\Omega^{(l-1)}_{\leq n}$ (we define $\Omega^{(0)}_n$ as X_n or its embedding).

Definition A.3. An *contextual* layer-wise decoder satisfies that for any two possible inputs $x_{<N}, x'_{<N}$ and $n < N$, if $p(X_{n+1}|x_{\leq n}) \neq p(X_{n+1}|x'_{\leq n})$, then $\omega^{(L)}_n \neq \omega'^{(L)}_n$, where $\omega^{(L)}_n$ ($\omega'^{(L)}_n$) is $\Omega^{(L)}_n$ evaluated on input $x_{\leq n}$ ($x'_{\leq n}$).

Our definition of contextual decoder aligns with the definition of contextual mapping in previous works (Yun et al., 2020; Kim et al., 2023), which guarantees that certain different inputs are mapped to different representations, although their definition of contextual mapping is more focused on the seq2seq setting.

Next, we make the following observations:

Proposition A.4. *For a layer-wise decoder on a data distribution, the prefixes of its intermediate representation at the l-th layer $\Omega^{(l)}_{\leq i}$ satisfy*

1. $H(X_{n+1}|\Omega^{(l)}_{\leq i}) \geq H(X_{n+1}|X_{\leq i})$ *for all* $i \leq n$;
2. $H(X_{n+1}|\Omega^{(l)}_{\leq n}) = H(X_{n+1}|X_{\leq n})$ *if the decoder is contextual;*
3. $H(X_{n+1}|\Omega^{(l)}_{\leq i} = \omega^{(l)}_{\leq i}) \geq H(X_{n+1}|X_{\leq i})$ *for all* $i \leq n$ *and all* $\omega^{(l)}_{\leq i}$ *if the data is causally isotropic;*
4. $H(X_{n+1}|\Omega^{(l)}_{\leq n} = \omega^{(l)}_{\leq n}) = H(X_{n+1}|X_{\leq n})$ *for all* $\omega^{(l)}_{\leq n}$ *if the decoder is contextual and the data is causally isotropic.*

Proof. 1. Notice that $X_{n+1} \to X_{\leq i} \to \Omega^{(l)}_{\leq i}$ is a Markov chain. By the data processing inequality (Polyanskiy & Wu, 2016), $I(\Omega^{(l)}_{\leq i}; X_{n+1}) \leq I(X_{\leq i}; X_{n+1}) \implies H(X_{n+1}|\Omega^{(l)}_{\leq i}) \geq H(X_{n+1}|X_{\leq i})$.

2. For any $\omega^{(l)}_{\leq n}$, let $\kappa(\omega^{(l)}_{\leq n})$ be the set of inputs where $p(x_{\leq n}|\omega^{(l)}_{\leq n}) > 0$, which is equivalent to $p(\omega^{(l)}_{\leq n}|x_{\leq n}) = 1$ by the deterministic nature of decoder.

   By the definition of contextual layer-wise decoder,

$$\begin{aligned} \forall x_{\leq n}, x'_{\leq n} \in \kappa(\omega^{(l)}_{\leq n}) &\implies \omega^{(l)}_{\leq n} = \omega'^{(l)}_{\leq n} \implies \omega^{(L)}_{\leq n} = \omega'^{(L)}_{\leq n} \\ &\implies \omega^{(L)}_n = \omega'^{(L)}_n \implies p(X_{n+1}|x_{\leq n}) = p(X_{n+1}|x'_{\leq n}). \end{aligned} \tag{12}$$

   Therefore,

$$\begin{aligned} p(X_{n+1}|\omega^{(l)}_{\leq n}) &= \sum_{x_{\leq n} \in \kappa(\omega^{(l)}_{\leq n})} p(X_{n+1}|x_{\leq n}, \omega^{(l)}_{\leq n}) p(x_{\leq n}|\omega^{(l)}_{\leq n}) \\ \text{(by conditional independence)} \quad &= \sum_{x_{\leq n} \in \kappa(\omega^{(l)}_{\leq n})} p(X_{n+1}|x_{\leq n}) p(x_{\leq n}|\omega^{(l)}_{\leq n}) \qquad (13) \\ &= p(X_{n+1}|x_{\leq n}),\ \forall x_{\leq n} \in \kappa(\omega^{(l)}_{\leq n}) \\ \implies H(X_{n+1}|\Omega^{(l)}_{\leq n} = \omega^{(l)}_{\leq n}) &= H(X_{n+1}|X_{\leq n} = x_{\leq n}),\ \forall x_{\leq n} \in \kappa(\omega^{(l)}_{\leq n}) \qquad (14) \end{aligned}$$

[4] Causal isotropy is a strict condition. We use it for demonstration purposes only: it isolates the effect of data variability in judging the disproportionality of attention.

$$\implies H(X_{n+1}|\Omega^{(l)}_{\leq n}) = \sum_{\omega^{(l)}_{\leq n}} p(\omega^{(l)}_{\leq n}) H(X_{n+1}|\Omega^{(l)}_{\leq n} = \omega^{(l)}_{\leq n}) = \sum_{\omega^{(l)}_{\leq n}} \left(\sum_{x_{\leq n} \in \kappa(\omega^{(l)}_{\leq n})} p(x_{\leq n}) \right) H(X_{n+1}|\Omega^{(l)}_{\leq n} = \omega^{(l)}_{\leq n})$$
$$= \sum_{x_{\leq n}} p(x_{\leq n}) H(X_{n+1}|X_{\leq n} = x_{\leq n}) = H(X_{n+1}|X_{\leq n}). \tag{15}$$

3. Note that (13) can be written as a weighted average, which we denote as avg_κ:

$$p(x_{n+1}|\omega^{(l)}_{\leq n}) = \text{avg}_\kappa p(x_{n+1}|x_{\leq n}),\ \forall x_{n+1}. \tag{16}$$

Similarly, with a slightly different definition of κ,

$$p(x_{n+1}|\omega^{(l)}_{\leq i}) = \text{avg}_\kappa p(x_{n+1}|x_{\leq i}), \forall x_{n+1}. \tag{17}$$

Apply Jensen's inequality to the function $-x \log x$, we have for any x_{n+1},

$$\begin{aligned} -p(x_{n+1}|\omega^{(l)}_{\leq i}) \log p(x_{n+1}|\omega^{(l)}_{\leq i}) &= -\left(\text{avg}_\kappa p(x_{n+1}|x_{\leq i})\right) \log \left(\text{avg}_\kappa p(x_{n+1}|x_{\leq i})\right) \\ &\geq \text{avg}_\kappa \left(-p(x_{n+1}|x_{\leq i}) \log p(x_{n+1}|x_{\leq i})\right). \end{aligned} \tag{18}$$

Therefore,

$$\begin{aligned} H(X_{n+1}|\Omega^{(l)}_{\leq i} = \omega^{(l)}_{\leq i}) &= -\sum_{x_{n+1}} p(x_{n+1}|\omega^{(l)}_{\leq i}) \log p(x_{n+1}|\omega^{(l)}_{\leq i}) \\ &\geq \sum_{x_{n+1}} \text{avg}_\kappa \left(-p(x_{n+1}|x_{\leq i}) \log p(x_{n+1}|x_{\leq i})\right) \\ &= \text{avg}_\kappa \sum_{x_{n+1}} -p(x_{n+1}|x_{\leq i}) \log p(x_{n+1}|x_{\leq i}) \\ &= \text{avg}_\kappa H(X_{n+1}|X_{\leq i} = x_{\leq i}) \\ \text{(by causal isotropy)} \quad &= H(X_{n+1}|X_{\leq i}). \end{aligned} \tag{19}$$

4. Apply causal isotropy to (14).

□

We are now ready to define the disproportionality of attention:

Definition A.5. Let inputs sampled from a data distribution $p(X_1, X_2, \ldots, X_N)$ run through a contextual layer-wise decoder with attention layers. If for at least one possible input $x_{<N}$, the attention $\tilde{A}^{(l)}$ after softmax in the l-th layer satisfy

$$\sum_{j \leq i} \tilde{A}^{(l)}_{nj} > \frac{I(X_{\leq i}; X_{n+1})}{I(X_{\leq n}; X_{n+1})} \sum_{j \leq n} \tilde{A}^{(l)}_{nj} + \varepsilon \tag{20}$$

for some $i < n < N$ and $I(X_{\leq n}; X_{n+1}) > 0$, then this attention layer is said to have disproportional attention towards initial tokens on this input. The overall degree of disproportionality of an attention layer can be measured by the total probability of such inputs $\sum_{x_{<N}} p(x_{<N})$.

Note that by Proposition A.4, the following always holds:

$$\frac{I(X_{\leq i}; X_{n+1})}{I(X_{\leq n}; X_{n+1})} = \frac{H(X_{n+1}) - H(X_{n+1}|X_{\leq i})}{H(X_{n+1}) - H(X_{n+1}|X_{\leq n})} \geq \frac{H(X_{n+1}) - H(X_{n+1}|\Omega^{(l)}_{\leq i})}{H(X_{n+1}) - H(X_{n+1}|\Omega^{(l)}_{\leq n})} = \frac{I(\Omega^{(l)}_{\leq i}; X_{n+1})}{I(\Omega^{(l)}_{\leq n}; X_{n+1})}. \tag{21}$$

This justifies our choice of the threshold $\frac{I(X_{\leq i}; X_{n+1})}{I(X_{\leq n}; X_{n+1})}$ for detecting the disproportionality of attention. Moreover, if the data is causally isotropic, the specific values of data do not matter for how much attention the model should pay.

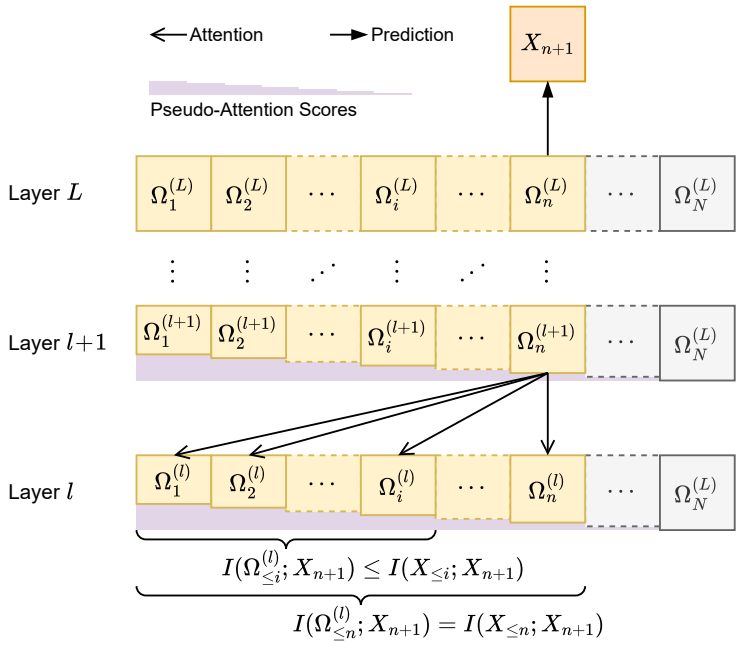

Figure 6. The DA issue and the proposed solution of adding pseudo-attention scores. The rationale behind is that through learning, a decoder should learn to avoid paying too much attention to where $H(X_{n+1}|\Omega^{(l)}_{\leq i} = \omega^{(l)}_{\leq i})$ is high, because such places provide little mutual information with respect to the prediction goal.

In this work, we handle the DA problem by pseudo-attention scores and we offer a probabilistic interpretation. First, we clarify that the problem does not lie in the query-key-value mechanism of attention, but rather lies in the nature of autoregression: the history does not represent a complete description of the future, and the probability that the future deviates from the history must be taken into account, and more so at the beginning. Thus the output of an attention layer at earlier positions should be able to signal to the subsequent layers a higher variance of estimation compared to later positions. The failure of reliably doing so leads to the model having to allocate computation elsewhere to rectify the signal, such as excessive attention towards irrelevant tokens (Xiao et al., 2023) and "no-op" heads (Bondarenko et al., 2023), or becoming totally paralyzed (Appendix B.1). StableMask parameterizes this inductive bias orthogonal to decoder-only Transformers with RPE by pseudo-attention scores in the causal mask that decays over time.

B. Further Explanation of Position Encoding

B.1. The Unit Test of Absolute Position-Awareness

Training a decoder-only Transformer with no PE will fail on data points that consist of all identical tokens, because the outputs of each layer are all identical vectors. Consequently, it is impossible for the model to predict different output distributions at different positions. We regard such all-identical inputs with different outputs at different positions as the "unit test" of absolute position awareness. We showed that Transformers with RPE cannot pass this test (Appendix B.2).

One way to pass the test without using explicit PE was proposed, by prepending a special $\langle bos \rangle$ token to the input sequence (Kazemnejad et al., 2023). It breaks the symmetry in all positions and provides a way for the decoder to recognize absolute position. We note that this solution is equivalent to the AT-based method used to solve the DA issue (Section 3). This inspires us to see the test from the viewpoint of DA. Indeed, we have

Theorem B.1. *There exists a causally isotropic data distribution (defined in Appendix A) such that any regular Transformer decoder has a high probability of being (weakly) disproportional in all of its attention layers.*

Proof. Consider the following $\mathrm{softCopyLast}$ task: for any input $x_{<n}$, output the last token with probability $1 - e^{-n}$, or a random token otherwise. The training dataset is constructed by a sampling algorithm that correctly does the task repeatedly.

The training dataset is causally isotropic: for every set of observed variables x_Λ, $\Lambda \subset [n]$, $H(X_{n+1}|X_\Lambda = x_\Lambda)$ depends only on the largest element of Λ, not on the specific values of variables.

Moreover, the probability density of this dataset concentrates most on the all-identical sequences, because as time goes on, sequences in the dataset are increasingly likely to copy themselves.

Last, we need to check that regular Transformer decoders have (weakly) disproportional attention on all identical sequences in all the attention layers. Note that although $I(X_{\leq i}; X_{n+1}) > 0$ for $i < n$, $I(X_{\leq i}; X_{n+1}|X_n) = 0$ holds because of conditional independence between $X_{\leq i}$ and X_{n+1} given X_n. On the other hand, $\sum_{j\leq i} \tilde{A}^{(l)}_{nj} > \varepsilon$ holds because the softmax in a regular Transformer always gives positive attention. So the model has a weak disproportional attention towards initial tokens. □

Intuitively speaking, if the inputs are all identical, then the model only needs to know the last token and the sequence length in order to decide the output. All other attention can be regarded as (weakly) disproportional. However, inputs constructed this way only account for an exponentially small total probability in real datasets, so we separate this issue from the issue of disproportional attention.

B.2. Experiment of RPE's Inability to Encode Absolute Position

To demonstrate that RPE cannot encode absolute positional information as discussed in Section 3, we designed several experiments that require knowledge of absolute positional relationships. These experiments primarily include three tasks:

(1) Absolute Position Mapping: Given an input sequence of "0 0 0 0 0 ...", the model needs to accurately map each position to its absolute position. In other words, we expect an output of "1 2 3 4 5 ...".

(2) Absolute Position Identification: Given an input sequence of "0 0 0 ... `[ABE]` 0 0 ...", where `[ABE]` encodes a special character at a specific position, the model needs to output the absolute position corresponding to the location encoded by `[ABE]`. In this case, we expect an output of "0 0 0 ... n 0 0 ...", where n represents the current position.

(3) Odd-Even Number Counting: Given an input sequence of "0 0 0 0 0 ...", the model needs to output a sequence of consecutive odd and even numbers, such as "1 2 1 2 ...". This task also relies on the model's ability to recognize absolute positional information.

	Accuracy		
PE*	Task (1)	Task (2)	Task (3)
APE			
Learnable	**96.7**%	**94.3**%	**97.6**%
Sinusoidal	**98.1**%	**99.1**%	**96.2**%
RPE			
ALiBi	21.7%	26.7%	46.5%
T5	22.4%	24.5%	42.7%
RoPE	25.3%	24.7%	43.1%

**: positional encoding type*

Table 4. Experiment settings and Results of RPE's inability to encode absolute position. We designed three datasets that rely on absolute position information and calculated average accuracy on these tasks. The results show that RPE performs poorly. This demonstrates that the position information encoded during the softmax process in RPE is shadowed.

Our experiments were conducted using a model with 160 million parameters, trained on four V100 GPUs. For detailed training hyperparameters, one can refer to the training details on the Wikitext-103 dataset (Appendix F).

C. StableMask Encodes Absolute Positional Information

In this section, we present how StableMask can recover absolute positions in the hidden state using fewer portions of the model, than prepending a special $\langle bos \rangle$ token to the sequence (Appendix B.1). Our proof is inspired by NoPE (Kazemnejad

et al., 2023) but differs substantially in that they require three dimensions of hidden states at free disposal, while ours only needs two and is arguably more natural.

Theorem C.1. *Let $X = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n]_n$ be an input sequence of length n to the StableMask model $f_T^{(SM)}$. Then, the first layer of $f_T^{(SM)}$ can recover absolute positions $[1, 2, \ldots, n]$ in the hidden state $\Omega^{(1)}$. That is, there exist W_Q, W_K, W_V and W_O for the first attention layer, along with W_1 and W_2 for the first feed-forward layer, that computes absolute positions and pass them to the next layer.*

Proof. We focus on the goal of reconstructing an index-dependent function $\xi_i = i/(i + \sum_{j=i}^{n-1} e^{-j\gamma})$ at the end of the first attention layer. After reconstructing ξ_i, recovering i from it can be done by the universal approximation power of feed-forward networks (Park et al., 2021).

For this, we need to gain control of a single head in the first attention layer, and use two hidden dimensions in the embedding layer. Note that this approach does not alter the rest of the Transformer model.

First, we specify the word embedding matrix $W_E \in \mathbb{R}^{d \times \mathcal{V}}$ as follows: the first row of W_E is set to 1, which serves as the input vector; The second row of W_E is set to 0, which serves as the output vector. Then, we have:

$$W_E = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ 0 & 0 & \ldots & 0 \\ e_{3,1} & e_{3,2} & \cdots & e_{3,\mathcal{V}} \\ \vdots & \vdots & \ddots & \vdots \\ e_{d,1} & e_{d,2} & \cdots & e_{d,\mathcal{V}} \end{pmatrix}_{d \times \mathcal{V}} \tag{22}$$

where $e_{i,j} \in \mathbb{R}$. The word embeddings for the input sequence $X = [x_1, \ldots, x_n]_n$ are retrieved from the embedding matrix W_E by:

$$X_E = W_E[X] = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ 0 & 0 & \ldots & 0 \\ e_{3,x_1} & e_{3,x_2} & \cdots & e_{3,x_n} \\ \vdots & \vdots & \ddots & \vdots \\ e_{d,x_1} & e_{d,x_2} & \cdots & e_{d,x_n} \end{pmatrix}_{d \times n} \tag{23}$$

Second, for head dimension $h \geq 1$, we specify the weights W_Q, W_K, W_V, W_O of the selected attention head in the first layer. Specifically, we set $W_Q = W_K = 0$, and

$$W_V = \begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 \end{pmatrix}_{h \times d}, \quad W_O = \begin{pmatrix} 0 & 0 & \ldots & 0 \\ 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 \end{pmatrix}_{d \times h}. \tag{24}$$

Consequently, all the query-key matching results are zero:

$$W_K X_E = W_Q X_E = 0_{h \times n}, \quad A = (W_Q X_E)^\top (W_K X_E) = 0_{n \times n}, \tag{25}$$

while W_V takes the first row of X_E, which is the input vector, and sets everywhere else zero:

$$W_V X_E = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 \end{pmatrix}_{h \times n} \tag{26}$$

We now calculate the output of attention. First, since the key-query matching results are all zero, the attention score matrix with StableMask is

$$A_{\text{SM}} = A \odot C + P = \begin{pmatrix} 0 & -\gamma & \cdots & -(n-1)\gamma \\ 0 & 0 & \cdots & -(n-1)\gamma \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}_{n \times n} \tag{27}$$

Therefore,

$$\tilde{A} = \mathrm{Softmax}(A_{\mathrm{SM}}) \odot C = \begin{pmatrix} 1/(1+\sum_{i=1}^{n-1} e^{-i\gamma}) & 0 & \cdots & 0 \\ 1/(2+\sum_{i=2}^{n-1} e^{-i\gamma}) & 1/(2+\sum_{i=2}^{n-1} e^{-i\gamma}) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1/n & 1/n & \cdots & 1/n \end{pmatrix}_{n\times n} \tag{28}$$

$$\tilde{A}(W_V X_E)^\top = \begin{pmatrix} 1/(1+\sum_{i=1}^{n-1} e^{-i\gamma}) & 0 & \cdots & 0 \\ 2/(2+\sum_{i=2}^{n-1} e^{-i\gamma}) & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{pmatrix}_{n\times h} = \begin{pmatrix} \xi_1 & 0 & \cdots & 0 \\ \xi_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \xi_n & 0 & \cdots & 0 \end{pmatrix}_{n\times h} \tag{29}$$

Finally, W_O is used to move the first row of $(\tilde{A}(W_V X_E)^\top)^\top$ to the second row:

$$W_O(\tilde{A}(W_V X_E)^\top)^\top = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ \xi_1 & \xi_2 & \cdots & \xi_n \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}_{d\times n} \tag{30}$$

Adding the residuals back to the input, we are done:

$$X_E + \sum_{\mathrm{h}} W_O^{(\mathrm{h})}(\tilde{A}^{(\mathrm{h})}(W_V^{(\mathrm{h})} X_E)^\top)^\top = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \xi_1 & \xi_2 & \cdots & \xi_n \\ * & * & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \cdots & * \end{pmatrix}_{d\times n} \tag{31}$$

where $*$ denotes values computed by other heads in the first layer, which we assumed to not interfere with the first two hidden dimensions. □

D. FlashAttention with StableMask

D.1. Introduction to FlashAttention

FlashAttention (Dao et al., 2022) is a state-of-the-art method designed to enhance the performance of attention mechanisms in Transformer models, particularly addressing the efficiency constraints imposed by modern GPU memory hierarchies. Traditional attention mechanisms suffer from significant computational overhead, predominantly due to the necessity of storing and accessing large intermediate matrices, such as the softmax-normalized attention scores, from the High Bandwidth Memory (HBM). This process is inherently memory-bound due to the quadratic dependency on the sequence length, leading to extensive memory accesses and thus increased wall-clock time.

The A100 GPU, for instance, showcases the discrepancy in memory speeds within its hierarchy, having a significantly faster on-chip SRAM compared to the larger HBM. FlashAttention optimizes for this architectural detail by reducing HBM reads and writes. It achieves a sub-quadratic number of HBM accesses by employing techniques like tiling and recomputation, which allow for the attention computation to be performed in smaller, more manageable blocks within the on-chip SRAM. This block-based approach mitigates the need to store large intermediate matrices, especially beneficial during the backward pass of model training where intermediate values are traditionally saved to HBM.

Furthermore, FlashAttention incorporates kernel fusion in its implementation, enabling a single CUDA kernel to handle the entire computation process – from loading inputs from HBM, through all the computation steps (such as matrix multiplication and softmax), to writing the results back to HBM. This minimizes the frequency of costly memory accesses and contributes to an overall faster computation, without compromising the accuracy of the attention mechanism. As a result, FlashAttention stands out as an efficient primitive for both memory-bound and compute-bound operations within the GPU's memory hierarchy, offering a significant improvement in the execution of Transformer models.

D.2. Derivation

In the FlashAttention paradigm, the query $Q \in \mathbb{R}^{n \times d_H}$, key $K \in \mathbb{R}^{n \times d_H}$, and value $V \in \mathbb{R}^{n \times d_H}$ matrices are partitioned into $Tr = \frac{n}{Br}$ blocks $Q_1, \ldots, Q_{Tr}$, $K_1, \ldots, K_{Tr}$, $V_1, \ldots, V_{Tr}$, each of dimension $\mathbb{R}^{Br \times d_H}$. Then each block Q_i, K_j, V_i is fetched for computation. The attention scores $S_i^{(j)}$ for blocks Q_i and K_j are derived from the on-chip computation: $S_i^{(j)} = Q_i K_j^T \in \mathbb{R}^{Br \times Br}$. With the incorporation of StableMask, two additional on-chip operations are introduced:

$$S_i^{(j)} = (Q_i K_j^T) \odot C_i^{(j)} + P_i^{(j)}, \tag{32}$$

where P and C correspond to the StableMask matrices, segmented into $Tr \times Tr$ blocks with $P_i^{(j)}, C_i^{(j)} \in \mathbb{R}^{Br \times Br}$, and loaded on-chip. The safe softmax operation, analogous to that in FlashAttention, proceeds as follows:

$$m_i^{(j)} = \max(m_i^{(j-1)}, \text{rowmax}(S_i^{(j)})) \in \mathbb{R}^{Br}, \tag{33}$$

$$\tilde{S_i^{(j)}} = \exp(S_i^{(j)} - m_i^{(j)}) \in \mathbb{R}^{Br \times Br}, \tag{34}$$

$$l_i^{(j)} = e^{m_i^{(j)} - m_i^{(j-1)}} l_i^{(j-1)} + \text{rowsum}(\tilde{S_i^{(j)}}) \in \mathbb{R}^{Br}. \tag{35}$$

Subsequently, the algorithm rectifies the attention score matrix to account for zeros necessitated by the causal mask, so the final output $O_i^{(j)}$ is computed as:

$$O_i^{(j)} = \text{diag}(e^{m_i^{(j)} - m_i^{(j-1)}})^{-1} O_i^{(j-1)} + (\tilde{S_i^{(j)}} \odot C_i^{(j)}) V_i. \tag{36}$$

D.3. A Typical Implementation of FlashAttention 2

Algorithm 1 Forward pass

Require: Matrices $\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{C}, \mathbf{B} \in \mathbb{R}^{N \times d}$ in HBM, block sizes B_c, B_r.

Divide $\mathbf{Q}$ into $T_r = \left\lceil \frac{N}{B_r} \right\rceil$ blocks $\mathbf{Q}_1, \ldots, \mathbf{Q}_{T_r}$ of size $B_r \times d$ each, and divide $\mathbf{K}, \mathbf{V}$ in to $T_c = \left\lceil \frac{N}{B_c} \right\rceil$ blocks $\mathbf{K}_1, \ldots, \mathbf{K}_{T_c}$ and $\mathbf{V}_1, \ldots, \mathbf{V}_{T_c}$, of size $B_c \times d$ each. Divide $\mathbf{C}, \mathbf{P}$ in to $T_r \times T_c = \left\lceil \frac{N}{B_r} \right\rceil \times \left\lceil \frac{N}{B_c} \right\rceil$ blocks $\mathbf{C}_1, \ldots, \mathbf{C}_{T_r}$ and $\mathbf{P}_1, \ldots, \mathbf{P}_{T_c}$, of size $B_r \times B_c$ each.
Divide the output $\mathbf{O} \in \mathbb{R}^{N \times d}$ into T_r blocks $\mathbf{O}_i, \ldots, \mathbf{O}_{T_r}$ of size $B_r \times d$ each, and divide the logsumexp L into T_r blocks $L_i, \ldots, L_{T_r}$ of size B_r each.
for $1 \le i \le T_r$ **do**
  Load $\mathbf{Q}_i$ from HBM to on-chip SRAM.
  On chip, initialize $\mathbf{O}_i^{(0)} = (0)_{B_r \times d} \in \mathbb{R}^{B_r \times d}, \ell_i^{(0)} = (0)_{B_r} \in \mathbb{R}^{B_r}, m_i^{(0)} = (-\infty)_{B_r} \in \mathbb{R}^{B_r}$.
  for $1 \le j \le T_c$ **do**
    Load $\mathbf{K}_j, \mathbf{V}_j$ $\mathbf{C}_i^{(j)}, \mathbf{P}_i^{(j)}$ from HBM to on-chip SRAM.
    On chip, compute $\mathbf{S}_i^{(j)} = \mathbf{Q}_i \mathbf{K}_j^T \odot \mathbf{C}_i^{(j)} + \mathbf{P}_i^{(j)} \in \mathbb{R}^{B_r \times B_c}$.
    On chip, compute $m_i^{(j)} = \max(m_i^{(j-1)}, \text{rowmax}(\mathbf{S}_i^{(j)})) \in \mathbb{R}^{B_r}$, $\tilde{\mathbf{P}}_i^{(j)} = \exp(\mathbf{S}_i^{(j)} - m_i^{(j)}) \in \mathbb{R}^{B_r \times B_c}$ (pointwise), $\ell_i^{(j)} = e^{m_i^{j-1} - m_i^{(j)}} \ell_i^{(j-1)} + \text{rowsum}(\tilde{\mathbf{P}}_i^{(j)}) \in \mathbb{R}^{B_r}$.
    On chip, compute $\tilde{\mathbf{D}}_i^{(j)} \tilde{\mathbf{P}}_i^{(j)} \odot \mathbf{C}_i$.
    On chip, compute $\mathbf{O}_i^{(j)} = \text{diag}(e^{m_i^{(j-1)} - m_i^{(j)}})^{-1} \mathbf{O}_i^{(j-1)} + \mathbf{D}_i^{(j)} \mathbf{V}_j$.
  end for
  On chip, compute $\mathbf{O}_i = \text{diag}(\ell_i^{(T_c)})^{-1} \mathbf{O}_i^{(T_c)}$.
  On chip, compute $L_i = m_i^{(T_c)} + \log(\ell_i^{(T_c)})$.
  Write $\mathbf{O}_i$ to HBM as the i-th block of $\mathbf{O}$.
  Write L_i to HBM as the i-th block of L.
end for
Return the output $\mathbf{O}$ and the logsumexp L.

(The parts that are different from the original algorithm are marked in purple.)

Normal γ		Small γ		Large γ	
γ	PPL	γ	PPL	γ	PPL
0.5	24.33	1e-2	25.01	5	25.91
1	24.34	1e-3	24.99	1e2	28.39
0.25	24.41	1e-4	25.12	2e2	29.11

Table 5. Experiment for selection of hyperparameter τ

E. Selection of Hyperparameters

Here we provide our searching experiment for choosing gamma on OpenLLaMA 160M. In order to determine the optimal γ, we conducted a search across various numerical ranges, including values less than 1e-2, values greater than 10, and values near 0. Through our search, we have identified the optimal gamma to be 0.5.

F. Training Details

Parameters	71M	160M	400M	1.4B
Embedding Size	512	768	1024	2048
Hidden Size (Attention)	512	1536	2048	4096
Hidden Size (FFN)	2048	3072	2048	8192
Expanding Rate (FFN)	4	4	2	4
Activation Function	SwishGeLU	SwishGeLU	SwishGeLU	SwishGeLU
Normalization Type	RMSNorm	RMSNorm	RMSNorm	RMSNorm
Positional Encoding	RoPE / ALiBi	RoPE / ALiBi	RoPE / ALiBi	RoPE
Tokenizer	GPT2 Tokenizer	GPT2 Tokenizer	GPT2 Tokenizer	GPT2 Tokenizer
Vocabulary Size	50257	50257	50257	50257
# of Attention Heads	8	12	16	16
# of Layers	6	12	16	24

Table 6. Hyperparameters for WikiText-103 with ALibi and RoPE positional encoding

Hyperparameters for Wikitext-103		Hyperparameters for MiniPile		Hyperparameters for the Pile	
Data	WikiText-103	Data	MiniPile	Data	Pile
Sequence Length	512	Sequence Length	512 / 1024	Sequence Length	1024
Batch Size	64	Batch Size	128	Batch Size	128
Tokens per Batch	32768	Tokens per Batch	65536 / 131072	Tokens per Batch	131072
Total Steps	50k	Steps per Epoch	22k	Total Steps	200k
Warmup Steps	4k	Total Epoch	2	Warmup Steps	4k
Beginning Learning Rate	1e-6	Warmup Steps	4k	Beginning Learning Rate	5e-6
Peak Learning Rate	6e-4	Beginning Learning Rate	1e-6	Peak Learning Rate	2e-4
Learning Rate Decay	Linear	Peak Learning Rate	4e-4	Learning Rate Decay	Cosine
Optimizer	AdamW	Learning Rate Decay	Linear	Optimizer	AdamW
Adam ϵ	1×10^{-8}	Optimizer	AdamW	Adam ϵ	1×10^{-8}
Adam β_1	0.9	Adam ϵ	1×10^{-8}	Adam β_1	0.9
Adam β_2	0.98	Adam β_1	0.9	Adam β_2	0.98
Hidden Dropout	0	Adam β_2	0.98	Hidden Dropout	0
GELU Dropout	0	Hidden Dropout	0	GELU Dropout	0
Attention Dropout (if needed)	0	GELU Dropout	0	Attention Dropout (if needed)	0
Weight Decay	0.01	Attention Dropout (if needed)	0	Weight Decay	0.1
Gradient Clipping Value	1	Weight Decay	0.1	Gradient Clipping Value	1
Head-wise γ	True	Gradient Clipping Value	1	Head-wise γ	True
γ Value	0.5	Head-wise γ	True	γ Value	0.5

Table 7. Hyperparameters for WikiText-103 with ALibi and RoPE positional encoding

G. Visualization of Attention Heads with StableMask