REGLA: Refining Gated Linear Attention

Peng Lu[1] **Ivan Kobyzev**[2 †] **Mehdi Rezagholizadeh**[3*] **Boxing Chen**[2] **Philippe Langlais**[1 †]
[1]DIRO, Université de Montréal [2]Huawei Noah's Ark Lab [3]Advanced Micro Devices, Inc.
peng.lu@umontreal.ca ivan.kobyzev@huawei.com mehdi.rezagholizadeh@amd.com
boxing.chen@huawei.com felipe@iro.umontreal.ca

Abstract

Recent advancements in Large Language Models (LLMs) have set themselves apart with their exceptional performance in complex language modelling tasks. However, these models are also known for their significant computational and storage requirements, primarily due to the quadratic computation complexity of softmax attention. To mitigate this issue, linear attention has been designed to reduce the quadratic space-time complexity that is inherent in standard transformers. In this work, we embarked on a comprehensive exploration of three key components that substantially impact the performance of the Gated Linear Attention module: feature maps, normalization, and the gating mechanism. We developed a feature mapping function to address some crucial issues that previous suggestions overlooked. Then we offered further rationale for the integration of normalization layers to stabilize the training process. Moreover, we explored the saturation phenomenon of the gating mechanism and augmented it with a refining module. We conducted extensive experiments and showed our architecture outperforms previous Gated Linear Attention mechanisms in extensive tasks including training from scratch and post-linearization with continual pre-training.

1 Introduction

In the rapidly evolving field of Natural Language Processing (NLP), Transformer models have emerged as a groundbreaking innovation. These models have demonstrated unparalleled success across a wide array of tasks, revolutionizing our approach to understanding and generating natural language. They have proven their mettle in analyzing intricate documents, executing professional writing, and performing sophisticated reasoning tasks, thereby setting new benchmarks in the realm of NLP (OpenAI, 2023; Touvron et al., 2023a,b; Jiang et al., 2024; Xie et al., 2024).

The cornerstone of these Transformer models is the softmax attention mechanism. This mechanism, an extension inspired by the attention mechanism employed in Recurrent Neural Network (RNN) systems, has played a pivotal role in the success of Transformer models (Bahdanau et al., 2015; Vaswani et al., 2017). The softmax attention has outperformed RNN models in terms of parallelizability and the stability of gradient propagation over time, making it a preferred choice for many NLP tasks.

However, the softmax attention mechanism is not without its challenges. It requires substantial computational resources and high memory usage, which can be a significant hurdle in practical applications. As the length of the input increases, the required computation grows quadratically. This growth restricts the context window size and complicates the deployment of these models in real-world scenarios (Kwon et al., 2023). In addition to the issue of computational complexity, several studies have highlighted the limited length extrapolation capability of self-attention-based models (Press et al., 2022; Ruoss et al., 2023). Specifically, transformer models tend to underperform during inference if the sequence length of the test data exceeds that of the training data. As an order-invariant encoding mechanism, the self-attention-based encoder heavily depends on Position Embeddings (PEs) to model input orders. However, these studies reveal that the inability of transformers to handle long sequences can be attributed to the limited length generalization ability of these position embedding methods (Press et al., 2022; Zhao et al., 2024; Wang et al., 2024). This finding underscores the need to explore alternative architectures to address the challenges associated with long-sequence processing.

Numerous studies have been conducted with the

†Corresponding author.
*Work conducted while at Huawei Noah's Ark Lab

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 2884–2898

aim of mitigating this drawback by introducing linear attention operator (Choromanski et al., 2021; Peng et al., 2021; Katharopoulos et al., 2020; Beltagy et al., 2020; Tay et al., 2020; Zhang et al., 2024; Nahshan et al., 2023; Chen et al., 2024; Kasai et al., 2021). Unfortunately, existing linear attention mechanisms frequently struggle to match the modeling quality of softmax attention. Some work introduce gating mechanisms to improve the performance of linear attention (Schlag et al., 2021; Mao, 2022; Yang et al., 2024a). In this work, we delve into the different components of the Gated Linear Attention mechanism with the goal of optimizing the training process while ensuring rapid inference.

Our contribution can be summarized as follows: First, we find that previous suggestions overlook some crucial aspects. We address the instability issue of feature mapping functions by proposing a normalized exponential solution. Additionally, we introduce a variance reduction scaling factor to enhance its performance. Then we revisit the normalization layer, emphasizing its role in stabilizing the training process. Finally, we investigate the saturation phenomenon of the Gating Mechanism and enhance it with a refining module. By integrating our findings, we propose a novel architecture that outperforms previous Gated Linear Attention mechanisms across various tasks.

2 Background

We first briefly revisit the linear attention. Our method is grounded on these works by analyzing the essential components of them.

2.1 Softmax Attention

The softmax attention (SA) is the key component of the state-of-the-art transformer architectures. Given a sequence of N query vectors $\{q_i\}$, which attend to M key and value vectors. The attention module aggregates the values with the normalized outputs of a softmax function (Vaswani et al., 2017):

$$\mathrm{SA}(\boldsymbol{q}_i, \{\boldsymbol{k}_j\}, \{\boldsymbol{v}_j\}) = \sum_j \frac{\exp(\boldsymbol{q}_i^\top \boldsymbol{k}_j/\sqrt{d})}{\sum_{j'} \exp(\boldsymbol{q}_i^\top \boldsymbol{k}_{j'}/\sqrt{d})} \boldsymbol{v}_j, \tag{1}$$

where $\boldsymbol{q}_i, \boldsymbol{k}_i, \boldsymbol{v}_i$ are d dimensional vectors. For a given input query $\boldsymbol{q}_i$, computing the attention necessitates time and space complexity of $O(M)$, leading to a memory footprint of $O(MN)$ for full N queries. This bottleneck makes attention-based LLMs difficult to scale in terms of context window size since growing input length not only substantially escalates GPU computation but also complicates the management of Key-Value (KV) cache, particularly for decoder-based LLMs (Kwon et al., 2023).

2.2 Linear Attention

Linear Attention (LA) (Katharopoulos et al., 2020; Choromanski et al., 2021; Peng et al., 2021; Zheng et al., 2022; Qin et al., 2022; Nahshan et al., 2023) exchanges the computation order by decomposing the softmax function with *randomized* or *learnable* feature functions. Eq.1 can then be rewritten as

$$\boldsymbol{h}_i = \frac{\sum_j \boldsymbol{v}_j \phi(\boldsymbol{k}_j)^\top \phi(\boldsymbol{q}_i)}{\sum_{j'} \phi(\boldsymbol{k}_{j'})^\top \phi(\boldsymbol{q}_i)}, \tag{2}$$

where $\phi : \mathbb{R}^d \to \mathbb{R}^m$ is a m dimensional feature mapping function. Such an order exchanging enables to avoid computing the attention matrix of size $\mathbb{R}^{N\times M}$ for the full sequence and reduces the time complexity to $O(N)$. Existing methods generally utilize different functions to approximate softmax kernels. For example, Choromanski et al. (2021) propose a positive Orthogonal Random features approach (Favor+) and Peng et al. (2021) leverages random Fourier features to approximate attention functions, Katharopoulos et al. (2020) adopt a learnable linear transformation with $1 + \mathrm{elu}(\cdot)$ activation as the feature map and Kasai et al. (2021) propose to use a learned ReLU function: $\phi(\boldsymbol{x}) = \mathrm{ReLU}(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})$ as the feature map.

Another benefit of this feature map-based attention is that Eq. 2 can be further regrouped as a linear recurrence formulation because of the associative property of the matrix product as:

$$\boldsymbol{S}_t = \boldsymbol{S}_{t-1} + \boldsymbol{v}_t \phi(\boldsymbol{k}_t)^\top, \tag{3}$$

$$\boldsymbol{c}_t = \boldsymbol{c}_{t-1} + \phi(\boldsymbol{k}_t), \tag{4}$$

$$\boldsymbol{h}_t = \frac{\boldsymbol{S}_t \phi(\boldsymbol{q}_t)}{\boldsymbol{c}_t^\top \phi(\boldsymbol{q}_t)}, \tag{5}$$

where $\boldsymbol{S}_t \in \mathbb{R}^{d\times m}$ is the recurrent state matrix and $\boldsymbol{c}_t \in \mathbb{R}^m$ is the normalization vector. This linear recurrence can be regarded as a variant of fast weight additive outer products (Schmidhuber,

1992; Schlag et al., 2021). These techniques concentrate on either estimating or modifying the softmax operator, thus maintaining its original characteristics. When contrasted with the softmax attention, these techniques frequently sacrifice performance for efficiency, typically leading to diminished task performance.

2.3 Linear Attention with Gating Mechanisms (GLA)

Instead of approximating self-attention rigorously, recent works focus on improving the hidden state representation by introducing different gating mechanisms (Peng et al., 2021; Schlag et al., 2021; Mao, 2022). Peng et al. (2021) propose to add a gated update rule to Linear Attention which is inspired by gated recurrent neural networks (Hochreiter and Schmidhuber, 1997; Cho et al., 2014; Chung et al., 2014) to forget distant input with a recency bias. The state updating rule is as follows:

$$\boldsymbol{S}_t = g_t \boldsymbol{S}_{t-1} + (1 - g_t)\boldsymbol{v}_t \phi(\boldsymbol{k}_t)^\top, \quad (6)$$

$$\boldsymbol{c}_t = g_t \boldsymbol{c}_{t-1} + (1 - g_t)\phi(\boldsymbol{k}_t), \quad (7)$$

where $g_t = \text{Sigmoid}(\boldsymbol{W}_g \boldsymbol{x}) \in \mathbb{R}$ is a function with learnable parameters $\boldsymbol{W}_g \in \mathbb{R}^{1\times d}$. Schlag et al. (2021) propose a way to improve the vanilla gating method as Fast Weight Programmer (Schmidhuber, 1992) to forget information related to the current write key:

$$\boldsymbol{S}_t = \boldsymbol{S}_{t-1} - g_t \boldsymbol{S}_{t-1}\phi(\boldsymbol{k}_t)\phi(\boldsymbol{k}_t)^\top + g_t \boldsymbol{v}_t \phi(\boldsymbol{k}_t)^\top. \quad (8)$$

Mao (2022) investigates various update rule configurations and proposes a fast decaying rule inspired by Ba et al. (2016) and removes feature maps. The update rule is as:

$$\boldsymbol{S}_t = \boldsymbol{G}_t \odot \boldsymbol{S}_{t-1} + \boldsymbol{v}_t \phi(\boldsymbol{k}_t)^\top, \quad (9)$$

$$\boldsymbol{G}_t = \sigma(\boldsymbol{W}_z \boldsymbol{x}_t + \boldsymbol{b}_z)\sigma(\boldsymbol{W}_f \boldsymbol{x}_t + \boldsymbol{b}_f)^\top, \quad (10)$$

where $\boldsymbol{W}_z \in \mathbb{R}^{d\times d}$, $\boldsymbol{W}_f \in \mathbb{R}^{m\times d}$, $\boldsymbol{b}_z \in \mathbb{R}^d$, $\boldsymbol{b}_f \in \mathbb{R}^m$ are trainable parameters, $\odot$ is Hadamard product, and σ is the Sigmoid function. This gated rule learns to output a gating matrix instead of a scalar, thus leading to a more fine-grained information control. This mechanism is also adopted in a recent work (Yang et al., 2024a) which develops a chunked parallel formulation for gated linear attention to achieve more hardware-friendly training for large-scale models. Pramanik et al. (2023) also utilize this fast decay rule and evaluate their recurrent linear transformer in reinforcement learning problems.

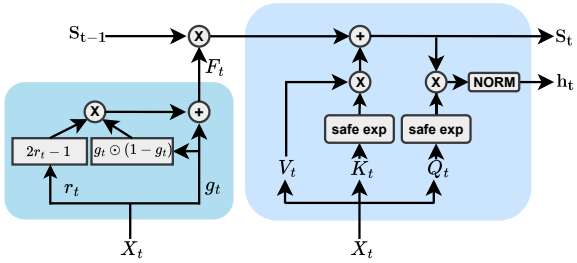

Figure 1: The overall model architecture of our ReGLA. The right side depicts the regular linear attention with our safe exp feature maps and normalization layer and the left side depicts the refining gate mechanism.

Compared to softmax attention's implicit unbounded memory footprint requirement: KV cache (Kwon et al., 2023), linear attention has bounded memory size during the inference, which is much easier to deploy and manage for language models in service. However, both the memory size of hidden states and the mechanism of updating rule have a great impact on the performance of these Linear models. For example, Schlag et al. (2021) develop the Deterministic Parameter-Free Projection (DPFP) to expand the outer product dimension and use delta rule to edit the forget/write mechanism of hidden states, but Mao (2022) demonstrates this underperforms the gating method. All these findings show that it's more crucial to concentrate on creating an expressive update rule for gate linear attention. It is not conclusive which architecture: softmax attention or linear Attention is superior. Also, techniques developed by efficient attention can be directly or indirectly adapted to various modern large language models to improve the deployment, i.e., Qin et al. (2023a) develop the first large-scale linear attention-based LLM and Slide Window Attention (SWA) (Beltagy et al., 2020) is reported being used in Mistral (Jiang et al., 2023) to achieve context extension for long input sequences.

3 Methodology

In this section, we develop a synergistic modification to the Gated Linear Attention via a comprehensive analysis of its three essential components: feature mapping, normalization layer and gating mechanism.

$\phi(\boldsymbol{z})$	Boundedness	Non-negativity
z	✗	✗
ReLU $= \max(\boldsymbol{z}, 0)$	✗	✓
FAVOR+	✓	✓
ELU$(\boldsymbol{z}) + 1$	✗	✓
$\{\cos(\boldsymbol{z}), \sin(\boldsymbol{z})\}$	✓	✗
$\exp(\boldsymbol{z} - \max(\boldsymbol{z}))$	✓	✓

Table 1: The boundedness and non-negativity characteristics of different feature maps, not even FAVOR+ hold the two essential properties, it requires redrawing random samples during training, thus introducing extra overhead.

3.1 Feature Maps

We start with the selection of feature maps. Few works focus on the forward computation stability of linear attention. We summarize the several commonly-used feature functions and analyze the boundedness and non-negativity of their corresponding inner product shown in Table 1.

Boundedness. We posit that the arbitrary value of the inner product of broadly feature map functions could induce training instability in the forward propagating, which cannot be addressed by adding a normalization layer (Qin et al., 2022) after the implicit inner product calculation. The issue comes from the unbounded value of the inner product of the features. We address this problem by using the normalized exponential feature mapping function. Assume that $x \in \mathbb{R}^{d \times L}$ - a sequence of vectors of length L and hidden size d. Define the corresponding query and key feature map as follows[1]:

$$\phi_q(\boldsymbol{x})_{i,l} = \exp((\boldsymbol{W}_q \boldsymbol{x})_{i,l} - \max_{1 \le j \le d}((\boldsymbol{W}_q \boldsymbol{x})_{j,l})), \tag{11}$$

$$\phi_k(\boldsymbol{x})_{i,l} = \exp((\boldsymbol{W}_k \boldsymbol{x})_{i,l} - \max_{\substack{1 \le j \le d \\ 1 \le s \le L}}((\boldsymbol{W}_k \boldsymbol{x})_{j,s})), \tag{12}$$

where $i \in [1, d]$ is the index of dimension and $l \in [1, L]$ is the order index of input element, $\boldsymbol{W}_q \in \mathbb{R}^{d \times d}$ and $\boldsymbol{W}_k \in \mathbb{R}^{d \times d}$ are learnable model parameters. Notice that the dot product of features is always bounded: $0 < \phi_q(\boldsymbol{x})^\top \phi_k(\boldsymbol{x}) \le d$, where d is the dimension of keys and queries. Indeed, each component of a vector of the form $\exp(z - \max(z))$ is bounded between 0 and 1, so the dot product can be upper-bounded by d and lower-bounded by 0.

Note that both Nahshan et al. (2023) and Zhang et al. (2024) choose to use exp functions but for different purposes: The first one estimates SA with log-normal distributions and the second one aims to retain the characteristics of original SA, including spikeness and monotonicity. Yet we select exp function from the perspective of training stability.

Variance Reduction Factor. Softmax attention applies a scaling factor $\frac{1}{\sqrt{d}}$ to the inner product ensuring stable model training by reducing the variance of the dot product to one (Vaswani et al., 2017). Previous linear attention works commonly follow the design and utilize the same scaling factor to the inner product $\phi_q(\boldsymbol{x})^\top \phi_k(\boldsymbol{x})$. However, we found that the variance of the inner product for linear attention not only depends on the feature dimension d but also related to the feature mapping functions. The following theorem provides the variance analysis of inner products with identity and exp functions.

Theorem 3.1 *Consider independent random variables $x_i, y_i \sim \mathcal{N}(0, 1)$, for $i \in [1, d]$. Define new variables u, z by:*

$$u = \sum_{i=1}^{d} x_i \times y_i, \tag{13}$$

$$z = \sum_{i=1}^{d} \exp(x_i) \times \exp(y_i). \tag{14}$$

Then the variance of u and z is d and $e^2(e^2 - 1)d$ respectively.

Based on the above theorem, we apply a new variance reduction factor $\frac{1}{e\sqrt{d(e^2-1)}}$ to the inner product in our linear attention to stabilize the training. We put the proof in the appendix. The left plot in Figure 2 shows the results of synthetic data. We randomly sampled 500 pairs of d dimensional vectors from the standard Gaussian distribution and computed the standard derivation of the inner product of them with two different feature mapping functions. The green and blue curve indicates the standard deviation given by the theorem 3.1 and we can see the two types of scatters almost completely follow the curves.

The right figure shows the standard derivation of real data. We sampled 100 inputs (each input has 1000 tokens) from the Wikitext-103 dataset and

[1]Note that there are alternative functions to replace $\max(\cdot)$. For example, $\log \sum \exp(\cdot)$ also ensures that the resulting inner-product remains bounded and this is equivalent to use a softmax function as the feature mapping.

	Model	Data	Tokens	Norm.	PPL
w/o Pretrain	**GLA**	WT	8M	✓	33.9
	GLA	WT	8M	✗	40.6
w/ Pretrain	**GLA**	SP	8M	✓	36.9
	GLA	SP	8M	✗	73.0

Table 2: We conducted experiments on two different datasets including Wikitext-103 (WT) and SlimPajama (SP) with and without pre-trained weights for different initialization.

input them to a one-layer linear attention model with two types of feature mapping functions and computed the corresponding standard deviation. The scatters show similar patterns: the variance is not only related to the input dimension but also depends on the feature map.

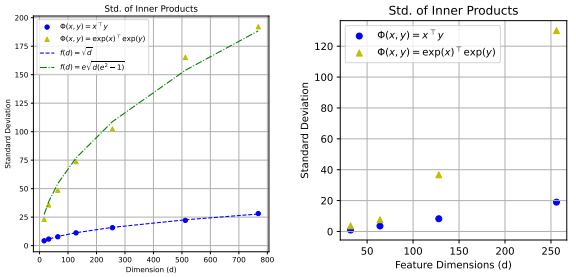

Figure 2: The left figure shows the standard derivation of synthetic data and the right figure shows the standard derivation of a one-layer linear attention with identical or exp feature map.

3.2 Normalization

Taxonomy of Normalization in LA There are two types of normalization terminology in the literature of linear attention: the first normalization comes from the denominator of Eq. 5 which corresponds to the original summation factor in the SA. We refer it to as *sum normalization*. Previous works show that it requires accumulation of positive values in Eq. 4, which may induce instability with growing inputs length (Schlag et al., 2021). Besides, some works found that the sum normalization can be dropped without performance degradation (Mao, 2022; Yang et al., 2024a). Apart from that, Qin et al. (2022) propose to add an extra normalization layer to address the unbounded gradient issues of different feature map functions we discussed above. We refer to it as *stable normalization*. There are many recent works adopting it to their methods (Qin et al., 2023a; Sun et al., 2023; Yang et al., 2024a; Mercat et al., 2024).

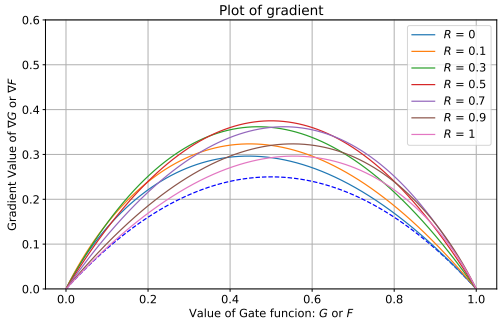

Figure 3: Gradient analysis of refined forgetting Gates and vanilla sigmoid gate. The dashed line indicates the gradient value of the vanilla sigmoid function $\nabla G = G \odot (1 - G)$. Other curves are the gradient of Refined forgetting gate ∇F in Eq. 16. It is a function of gating activation G_t and R_t. For activation values close to the boundary regions, the refined forget gate F has a higher gradient than the vanilla sigmoid function.

One follow-up *research question* is whether linear attention still needs a stable normalization layer when its feature maps are able to ensure bounded inner products. We conducted preliminary experiments by ablating the stable normalization layer after applying our feature mapping function. Unfortunately, we found performance degradation occurs for language modeling tasks as shown in Table 2. These results imply that the normalization layer not only helps to restrict the gradient of feature functions but also has other responsibilities to facilitate the training of linear attention. We conjecture the reason is that the variance of $\boldsymbol{h}_t$ is dependent on the input length t, especially after dropping the sum normalization.

3.3 Refined Gating Mechanism

Gated Linear Attention computes a weighted sum of the history information and a function of the current KV outer product to make the next hidden state. This update rule can be regarded as a residual connection (He et al., 2016). However, a widely discussed issue of gating mechanism in the literature of recurrent neural networks (Hochreiter and Schmidhuber, 1997; Cho et al., 2014; Lu et al., 2019; Chung et al., 2014; Gu et al., 2020) is the saturation problem of the sigmoid function $\boldsymbol{g} = \sigma(\boldsymbol{W}\boldsymbol{x})$, which is not explored for gated linear attention. The root reason is that the derivative of the sigmoid function could result in vanishing gradients around saturated regions: $\nabla \boldsymbol{g} = \boldsymbol{g} \odot (1 - \boldsymbol{g})$. Therefore, once the activation function $\boldsymbol{g}$ surpasses a certain upper or lower limit, the gradient vanishes

Method	Feature	Sum/Stable	Update Rule
LA (Katharopoulos et al., 2020)	ELU(x) + 1	✓/✗	$\boldsymbol{S}_t = \boldsymbol{S}_{t-1} + \boldsymbol{v}_t\phi(\boldsymbol{k}_t)^\top, \boldsymbol{c}_t = \boldsymbol{c}_{t-1} + \phi(\boldsymbol{k}_t),$
RFA (Peng et al., 2021)	$\cos(x), \sin(x)$	✓/✗	$\boldsymbol{S}_t = g_t\boldsymbol{S}_{t-1} + (1-g_t)\boldsymbol{v}_t\phi(\boldsymbol{k}_t)^\top, \boldsymbol{c}_t = g_t\boldsymbol{c}_{t-1} + (1-g_t)\phi(\boldsymbol{k}_t)$
DeltaNet (Schlag et al., 2021)	DPFP	✓/✗	$\boldsymbol{S}_t = \boldsymbol{S}_{t-1} - g_t\boldsymbol{S}_{t-1}\phi'(\boldsymbol{k}_t)\phi'(\boldsymbol{k}_t) + g_t\boldsymbol{v}_t\phi'(\boldsymbol{k}_t)^\top$
Fast Decay (Mao, 2022)	Identity	✗/✓	$\boldsymbol{S}_t = \boldsymbol{G}_t \odot \boldsymbol{S}_{t-1} + \boldsymbol{v}_t\phi(\boldsymbol{k}_t)^\top, \boldsymbol{G}_t = \boldsymbol{g}_z\boldsymbol{g}_f^\top$
REGLA	$\exp(\boldsymbol{x} - \max(\boldsymbol{x}))$	✗/✓	$\boldsymbol{S}_t = \boldsymbol{F}_t \odot \boldsymbol{S}_{t-1} + \boldsymbol{v}_t\phi(\boldsymbol{k}_t)^\top, \boldsymbol{F}_t = \left((\mathbf{1}-\boldsymbol{r}_t)\odot\boldsymbol{g}_t^2 + \boldsymbol{r}_t\odot(\mathbf{1}-(\mathbf{1}-\boldsymbol{g}_t)^2)\right)\mathbf{1}^\top,$

Table 3: Linear attention formulation with different feature maps, sum/stable normalization and updating rules.

rapidly, which prevents the learning of the gated representation.

We propose an enhancement to the gated linear attention mechanism via a refined gating mechanism designed to optimize the training process (Gu et al., 2020), particularly when the gate activation approaches saturation values. This is achieved by modifying the forget gate with an additional refining gate, thereby improving the overall performance and stability of the model. The updating rule is as follows:

$$\boldsymbol{S}_t = \boldsymbol{F}_t \odot \boldsymbol{S}_{t-1} + \boldsymbol{v}_t\phi(\boldsymbol{k}_t)^\top, \tag{15}$$

$$\boldsymbol{F}_t = \left((\mathbf{1}-\boldsymbol{r}_t)\odot\boldsymbol{g}_t^2 + \boldsymbol{r}_t\odot(\mathbf{1}-(\mathbf{1}-\boldsymbol{g}_t)^2)\right)\mathbf{1}^\top, \tag{16}$$

$$\boldsymbol{g}_t = \sigma(\boldsymbol{W}_g\boldsymbol{x} + \boldsymbol{b}_g), \tag{17}$$

$$\boldsymbol{r}_t = \sigma(\boldsymbol{W}_r\boldsymbol{x} + \boldsymbol{b}_r), \tag{18}$$

where $\odot$ is the Hadamard product, $\boldsymbol{W}_g \in \mathbb{R}^{d\times d}$, $\boldsymbol{W}_r \in \mathbb{R}^{d\times d}$, $\boldsymbol{b}_g \in \mathbb{R}^d$, $\boldsymbol{b}_r \in \mathbb{R}^d$ are trainable parameters. Eq. 16 calculates the gating activation $\boldsymbol{F}_t \in \mathbb{R}^{d\times d}$ and follows the outer-product gate form of (Mao, 2022; Yang et al., 2024a). We leverage a refining gate which was used to boost the performance of LSTM (Gu et al., 2020) by improving the gradient flow of the gating mechanism. The refining gate $\boldsymbol{r}_t$ interpolates between the lower band $\boldsymbol{g}_t^2$ and upper bound $\mathbf{1}-(\mathbf{1}-\boldsymbol{g}_t)^2$, which allows the gate activation $\boldsymbol{F}_t$ have a more effective activation range around the saturation region while keeping the value of F_t between 0 and 1. Figure 1 depicts the overall architecture of our gated linear attention with refining (REGLA). We present the gradient analysis in Figure 3. Notably, for activation values that are close to the boundary regions, the refined forget gate F exhibits a higher gradient than the standard sigmoid function.

4 Experiments

In this section, we evaluate our method with other linear attention and the conventional transformer. This comparison spans autoregressive language modeling training from scratch and finetuning pretrained language models after replacing its softmax attention with linear variants. To justify our design choices for REGLA, we conduct a comprehensive ablation study and efficiency analysis.

	Model	PPL
	Transformer	18.5
w/o Pretrain	LA w/ ReLU	28.5
	LA w/ ELU	31.3
	HedgeHog	22.4
	LA w/ Fast Decay	20.8
	REGLA(ours)	<u>19.0</u>
	Hybrid REGLA(ours)	**17.8**
w/ Pretrain	LA w/ ReLU	22.3
	LA w/ ELU	23.5
	HedgeHog	18.4
	LA w/ Fast Decay	18.2
	REGLA(ours)	<u>16.4</u>
	Hybrid REGLA(ours)	**14.8**

Table 4: Perplexity (PPL) of different linear attention configurations on the WikiText-103 test set. All Baselines use the same feature dimension 64 and for the training stability for all feature map functions, we apply stable normalization to the hidden representation.

4.1 Causal Language Modeling

Following previous work (Schlag et al., 2021; Kasai et al., 2021; Mao, 2022), we initially focus on autoregressive language modeling tasks and evaluate different methods on the Wikitext-103 dataset (Merity et al., 2017). For each method, we train a 160M parameter model for 50k steps with learning rate 2e-4, weight decay 0.01 and AdamW optimizer. For close comparison, we follow the architectural details of Pythia-160M (Biderman et al., 2023) with sequential connection and full RoPE embedding layer (Su et al., 2024), more specifically, it is a 12-layer decoder-only network with 12 heads, head dimension = 64, hidden dimension 768, and MLP dimension 3072. We compare various linear attention methods with different feature maps and updating rules.

Results. Table 4 shows the results of different methods on the WikiText-103 datasets. Among the models without pre-training, all methods based on linear attention still lag behind the Transformer

	Method	BoolQ	PIQA	HellaSwag	Winogrande	Truth_QA1	Truth_Qa2	Avg.
0-shot	**Pythia-160m**	54.6	62.0	**30.1**	**51.0**	24.9	44.5	44.5
	ReLU	55.5	56.5	26.6	48.6	23.5	47.2	43.0
	Hedgehog	60.5	**60.4**	27.7	50.2	24.4	46.0	44.9
	Scalar Gate	55.5	56.5	26.6	48.6	23.5	46.2	42.8
	Fast Decay	58.7	59.6	27.1	50.1	25.2	48.4	44.9
	REGLA	**62.0**	58.9	26.9	50.0	**25.3**	**48.8**	**45.3**
5-shot	**Pythia-160m**	50.6	**62.4**	30.7	51.4	24.9	44.5	44.1
	ReLU	56.5	58.4	26.0	50.2	24.2	45.5	43.5
	Hedgehog	61.4	55.6	27.0	50.8	25.7	49.6	45.0
	Scalar Gate	57.7	59.8	26.8	**51.8**	**26.4**	**50.1**	45.4
	Fast Decay	58.7	60.6	27.1	51.0	25.3	49.5	45.4
	REGLA	**62.1**	60.5	26.8	50.8	25.3	48.8	**45.7**

Table 5: Results of zero-shot and few-shot evaluation of Post-linearized Pythia-160m models.

models. However, our Refining Gated Linear Attention (REGLA) method significantly narrows this performance gap when compared to other methods, both with and without gating. This underscores the effectiveness of our design. We also implemented a hybrid architecture that mixes softmax attention layers with our REGLA layers. In our experiments, the replacement is conducted in a layer-wise manner. Specifically, for post-linearization, we replace 50% softmax attention layers (6 out of 12) in a Pythia-160m model with randomly initialized ReGLA modules and do continual training, for training from scratch, the architecture is the same, but both softmax attention and ReGLA modules are randomly initialized. We found this hybrid variant of REGLA outperforms the softmax attention method.

In addition to the aforementioned experiments, we also conducted continual pretraining experiments using pre-trained model checkpoints on WikiText. These experiments were carried out in a setting that aligns with those described in previous studies (Kasai et al., 2021; Mao, 2022). Specifically, we replaced the softmax attention of the Pythia-160m model with different linear attention mechanisms and applied continual pre-training to the entire model on the WikiText-103 dataset.

Our results underscore the versatility of our overall design. Not only is it effective when learning from scratch, but it also offers benefits for post-hoc linearization. This demonstrates the potential of our approach to enhance the performance of swapping existing SA models to their linear variants through continual pretraining.

We further evaluate the zero-shot and few-shot ability of the post-linearized models on common sense reasoning tasks, including BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), Winogrande (Sakaguchi et al., 2021), TruthfulQA 1 and 2 (Lin et al., 2022). The checkpoint of Pythia model is obtained from HuggingFace[2] and we use lm-evaluation-harness tool (Gao et al., 2023) to perform the 0-shot and 5-shot evaluation[3]. Since our REGLA also shares the outer product gating formulation as GLA (Yang et al., 2024a), we implemented it based on the Flash Linear Attention[4]. We replace the softmax attention layer with our method and other variants of linear attention. To recover the performance of the pre-trained model, we perform continual pre-training to the post-linearized model on the SlimPajama dataset (Soboleva et al., 2023) 50k steps with batch size 8 and maximum input length 2048.

Results. Table 5 presents the performance of various methods across six commonsense reasoning datasets. Following continual pretraining, our model effectively narrows the performance gap on most benchmarks, with PIQA and Hellaswag being the notable exceptions. Furthermore, our approach outperforms all baseline methods on average, demonstrating its superior performance in commonsense reasoning tasks.

5 Analysis and Discussion

In this section, we delve into a comprehensive discussion of our REGLA method. This includes an evaluation of the effectiveness of the gating mechanism, an analysis of speed and memory usage and an ablation study to understand the impact of each component. All of these aspects are examined in a controlled manner to ensure the reliability of our

[2]https://huggingface.co/EleutherAI/pythia-160m
[3]https://github.com/EleutherAI/lm-evaluation-harness
[4]https://github.com/sustcsonglin/flash-linear-attention

findings.

5.1 Gating Analysis

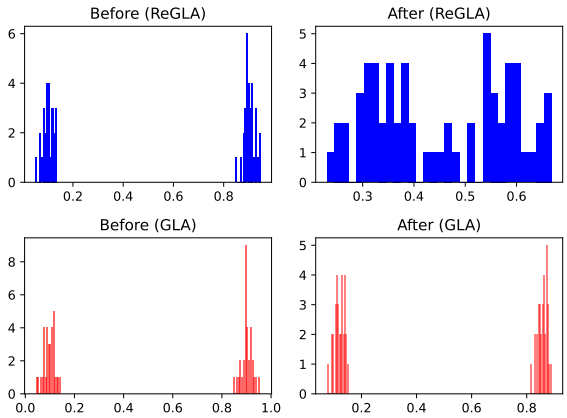

Figure 4: Distribution of gate activations before and after the training. We initialize the gate function with large and small biases to push two methods have very extreme gate activation values.

In addition to the aforementioned evaluations, we also conducted a detailed analysis of our refining gate mechanism. As depicted in Figure 4, we examined the distributions of the forget gate activations for both the Gated Linear Attention (GLA) and our Refining Gated Linear Attention (REGLA) methods, both before and after the training process.

To validate the effectiveness of our refining gate, we initialized the gate function with extremely large and small biases. This was done to push the initial activation values close to the boundary. The distribution after training revealed that the vanilla gating found it challenging to escape the extreme region. In contrast, our refined gate was able to learn a diverse range of activation distributions. Besides, we observed that the gate tended to concentrate on values significantly different from 1.0. This observation suggests that the language model may have a propensity to favor local information.

5.2 Memory and Speed Analysis

Next, we give an analysis of the inference speed and peak memory usage of our Refining Gated Linear Attention (REGLA) mechanism, comparing it with other methods, notably the Gated Linear Attention (GLA) with Fast Decay rule and softmax attention. Our experiments were conducted using 6-layer architectures. To ensure a more realistic comparison, we employed a Key-Value (KV) cache for softmax attention. All our experiments were carried out on a Nvidia V100 32GB GPU.

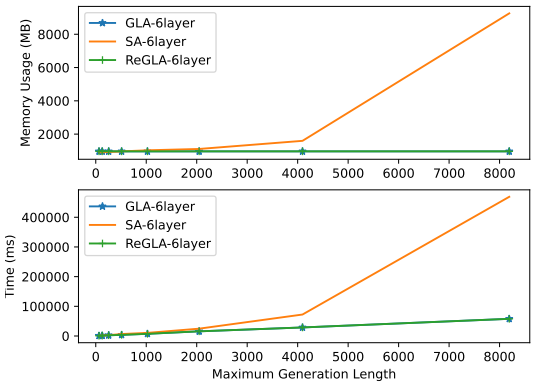

Figure 5: Plot of memory usage and the total prompt processing + decoding time of our REGLA, Fast Decay (GLA) and softmax attention (6-layer) when generating the next token at various sequence lengths on Nvidia V100 GPU. Our method and Fast Decay rule consume approximately the same peak memory and time (overlapped in plot).

We maintained a consistent prompt length of 5 and controlled the maximum generation length from 2^6 to 2^{13}. Figure 5 shows that softmax attention significantly consumes GPU memory as the output length increases, leading to a substantial slowdown in speed. In contrast, our REGLA, when compared to the Fast Decay rule, achieves nearly the same speed and memory footprints, demonstrating its efficiency and practicality.

Method	**Features config.**	PPL
REGLA	**16**	36.5
	32	24.7
	64	19.0
	96	18.8
	ReLU	21.5
	ELU + 1	23.7
	exp w/ $1/\sqrt{d}$	20.7
	exp w/ $1/e\sqrt{d(e^2-1)}$	19.0

Table 6: Ablation of different numbers of features and feature mappings in REGLA.

5.3 Ablation Study

We experimented with four distinct feature sizes and conducted these tests on the WikiText dataset. As indicated in Table 6, our observations reveal a clear trend of performance enhancement correlating with an increase in feature sizes. Apart from that, we analyze the effect of different feature mappings on REGLA and the impact of scaling factors, the results show the effectiveness of our exp func-

tion and the necessity of variance reduction scaling factor.

6 Related Work

There has been a great surge of research to design efficient variants of softmax attention or propose other alternatives for sequence modeling directly. The efficient variants broadly include two categories: *sparsified attention* and *linear attention*. Sparsified Attention (Beltagy et al., 2020; Zaheer et al., 2020; Tay et al., 2020; Kitaev et al., 2020) computes attention maps with pre-defined or learnable masks. For instance, Slide Window Attention (SWA) (Beltagy et al., 2020) limits each query input only attend to a certain number of preceding tokens. Another efficient variant is linear Attention (Katharopoulos et al., 2020; Choromanski et al., 2021; Peng et al., 2021; Zheng et al., 2022; Qin et al., 2022; Nahshan et al., 2023), which exchanges the computation order by decomposing the softmax function with *randomized* or *learnable* feature functions.

Alternatively, Gu et al. (2022) propose modeling sequential data with state-space models (SSMs) and show surprisingly good performance on a benchmark for comparing Transformers over long sequence data (Tay et al., 2021; Lu et al., 2023). The H3 model (Fu et al., 2023) expanded SSMs with gated connections and a conventional local convolution layer. They also show SSM can work in tandem with attention mechanism in a hybrid manner. Poli et al. (2023) propose to substitute the SSM layer with a global convolution parameterized by MLP. Gu and Dao (2023) incorporates data-dependent gating to SSMs and show comparable performance as transformer-based language models. Peng et al. (2023) develops RWKV architecture which absorbs insights from RNN and Attention-free transformer (Zhai et al., 2021). The RetNet model (Sun et al., 2023) and Transformer-LLM (Qin et al., 2023b) apply a decay factor to the current hidden state before incorporating the current input information and achieving impressive improvements. Yang et al. (2024a) and Yang et al. (2024b) develop chunkwise forms of GLA and DeltaNet respectively to parallelize the computation of gated linear recurrence models and provide a triton-based library to accelerate the training speed of linear attention model (Tillet et al., 2019).

Another interesting line of work dedicated to substituting the softmax attention in a pre-trained model with linear attention and performing continual training to bridge their performance gap. Kasai et al. (2021) take a pre-trained SA transformer, swap the SA modules with linear Attention, and continue training the entire model on the same task. Mao (2022) adopts the same procedure by optimizing it with the fast decay rules and removing the ReLU function in the feature maps, namely, a simple identity map. Chen et al. (2024); Mercat et al. (2024) linearized existing large pre-trained transformers into Recurrent Neural Networks (RNNs) with a modest continual pre-training budget to recover their performance. Wang et al. improve hybrid models by applying knowledge distillation (Hinton et al., 2015; Lu et al., 2021) from pre-trained transformers to mamba, enhancing efficiency and inference speed.

7 Conclusion

In this study, we conduct an in-depth examination of three pivotal components that significantly influence the performance of the Gated Linear Attention mechanism: Feature Maps, Normalization, and the Gating Mechanism. We posit the unstable issue of commonly used feature mapping functions and develop stable exponential functions. Apart from that, we also provide a corresponding variance reduction scaling factor to further improve its performance. Then we revisit the normalization layer and give additional justification for the incorporation of normalization layers to stabilize the training process. Furthermore, we explore the saturation phenomenon of the Gating Mechanism and enhance it with a refining mechanism. By integrating our findings, we propose a novel architecture that surpasses the performance of previous Gated Linear Attention mechanisms in extensive tasks.

Limitations

In this study, our primary focus is on autoregressive tasks. We believe that a concentrated examination of these tasks allows us to delve deeper into the nuances and intricacies involved, thereby providing more insightful and meaningful findings. Furthermore, our method is designed to investigate the fundamental components of linear attention methods. We aim to understand the underlying principles and mechanisms that drive the performance of these architectures. This approach allows us to identify potential areas for improvement and propose innovative solutions to enhance their ef-

fectiveness. We have not conducted large-scale experiments in this study. Our decision to limit the scale of our experiments is intentional. We believe that by focusing on a smaller, more manageable scale, we can maintain a high level of control and precision in our experiments. This approach ensures the reliability of our results and allows us to draw more accurate conclusions.

References

Jimmy Ba, Geoffrey E. Hinton, Volodymyr Mnih, Joel Z. Leibo, and Catalin Ionescu. 2016. Using fast weights to attend to the recent past. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4331–4339.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press.

Hanting Chen, Zhicheng Liu, Xutao Wang, Yuchuan Tian, and Yunhe Wang. 2024. Dijiang: Efficient large language models through compact kernelization. *CoRR*, abs/2403.19928.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 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 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J. Colwell, and Adrian Weller. 2021. Rethinking attention with performers. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2924–2936. Association for Computational Linguistics.

Herbert A. David and Haikady N. Nagaraja. 2005. Order statistics, third edition. In *Wiley Series in Probability and Statistics*.

Daniel Y. Fu, Tri Dao, Khaled Kamal Saab, Armin W. Thomas, Atri Rudra, and Christopher Ré. 2023. Hungry hungry hippos: Towards language modeling with state space models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *CoRR*, abs/2312.00752.

Albert Gu, Karan Goel, and Christopher Ré. 2022. Efficiently modeling long sequences with structured state spaces. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Albert Gu, Çaglar Gülçehre, Thomas Paine, Matt Hoffman, and Razvan Pascanu. 2020. Improving the

gating mechanism of recurrent neural networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3800–3809. PMLR.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

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

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *CoRR*, abs/2401.04088.

Jungo Kasai, Hao Peng, Yizhe Zhang, Dani Yogatama, Gabriel Ilharco, Nikolaos Pappas, Yi Mao, Weizhu Chen, and Noah A. Smith. 2021. Finetuning pretrained transformers into rnns. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 10630–10643. Association for Computational Linguistics.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5156–5165. PMLR.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*, pages 611–626. ACM.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3214–3252. Association for Computational Linguistics.

Peng Lu, Ting Bai, and Philippe Langlais. 2019. SC-LSTM: learning task-specific representations in multi-task learning for sequence labeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2396–2406. Association for Computational Linguistics.

Peng Lu, Abbas Ghaddar, Ahmad Rashid, Mehdi Rezagholizadeh, Ali Ghodsi, and Philippe Langlais. 2021. RW-KD: sample-wise loss terms re-weighting for knowledge distillation. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 3145–3152. Association for Computational Linguistics.

Peng Lu, Suyuchen Wang, Mehdi Rezagholizadeh, Bang Liu, and Ivan Kobyzev. 2023. Efficient classification of long documents via state-space models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6559–6565. Association for Computational Linguistics.

Huanru Henry Mao. 2022. Fine-tuning pre-trained transformers into decaying fast weights. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 10236–10242. Association for Computational Linguistics.

Jean Mercat, Igor Vasiljevic, Sedrick Keh, Kushal Arora, Achal Dave, Adrien Gaidon, and Thomas Kollar. 2024. Linearizing large language models. *arXiv preprint arXiv:2405.06640*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Yury Nahshan, Joseph Kampeas, and Emir Haleva. 2023. Linear log-normal attention with unbiased concentration. *CoRR*, abs/2311.13541.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Kiran GV, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartlomiej Koptyra, Hayden Lau, Jiaju Lin, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Johan S. Wind, Stanislaw Wozniak, Zhenyuan Zhang, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. 2023. RWKV: reinventing rnns for the transformer era. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 14048–14077. Association for Computational Linguistics.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A. Smith, and Lingpeng Kong. 2021. Random feature attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y. Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. 2023. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 28043–28078. PMLR.

Subhojeet Pramanik, Esraa Elelimy, Marlos C. Machado, and Adam White. 2023. Recurrent linear transformers. *CoRR*, abs/2310.15719.

Ofir Press, Noah A. Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Zhen Qin, Xiaodong Han, Weixuan Sun, Dongxu Li, Lingpeng Kong, Nick Barnes, and Yiran Zhong. 2022. The devil in linear transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 7025–7041. Association for Computational Linguistics.

Zhen Qin, Dong Li, Weigao Sun, Weixuan Sun, Xuyang Shen, Xiaodong Han, Yunshen Wei, Baohong Lv, Xiao Luo, Yu Qiao, et al. 2023a. Transnormerllm: A faster and better large language model with improved transnormer.

Zhen Qin, Dong Li, Weigao Sun, Weixuan Sun, Xuyang Shen, Xiaodong Han, Yunshen Wei, Baohong Lv, Xiao Luo, Yu Qiao, et al. 2023b. Transnormerllm: A faster and better large language model with improved transnormer.

Anian Ruoss, Grégoire Delétang, Tim Genewein, Jordi Grau-Moya, Róbert Csordás, Mehdi Bennani, Shane Legg, and Joel Veness. 2023. Randomized positional encodings boost length generalization of transformers. *CoRR*, abs/2305.16843.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. 2021. Linear transformers are secretly fast weight programmers. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 9355–9366. PMLR.

Jürgen Schmidhuber. 1992. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Comput.*, 4(1):131–139.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. 2023. Retentive network: A successor to transformer for large language models. *CoRR*, abs/2307.08621.

Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. 2020. Sparse sinkhorn attention. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9438–9447. PMLR.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2021. Long range arena : A benchmark for efficient transformers. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Philippe Tillet, Hsiang-Tsung Kung, and David D. Cox. 2019. Triton: an intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages, MAPL@PLDI 2019, Phoenix, AZ, USA, June 22, 2019*, pages 10–19. ACM.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Junxiong Wang, Daniele Paliotta, Avner May, Alexander M Rush, and Tri Dao. The mamba in the llama: Distilling and accelerating hybrid models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Suyuchen Wang, Ivan Kobyzev, Peng Lu, Mehdi Rezagholizadeh, and Bang Liu. 2024. Resonance rope: Improving context length generalization of large language models. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 586–598. Association for Computational Linguistics.

Qianqian Xie, Dong Li, Mengxi Xiao, Zihao Jiang, Ruoyu Xiang, Xiao Zhang, Zhengyu Chen, Yueru He, Weiguang Han, Yuzhe Yang, Shunian Chen, Yifei Zhang, Lihang Shen, Daniel S. Kim, Zhiwei Liu, Zheheng Luo, Yangyang Yu, Yupeng Cao, Zhiyang Deng, Zhiyuan Yao, Haohang Li, Duanyu Feng, Yongfu Dai, VijayaSai Somasundaram, Peng Lu, Yilun Zhao, Yitao Long, Guojun Xiong, Kaleb Smith, Honghai Yu, Yanzhao Lai, Min Peng, Jianyun Nie, Jordan W. Suchow, Xiao-Yang Liu, Benyou Wang, Alejandro Lopez-Lira, Jimin Huang, and Sophia Ananiadou. 2024. Open-finllms: Open multimodal large language models for financial applications. *CoRR*, abs/2408.11878.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. 2024a. Gated linear attention transformers with hardware-efficient training. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. 2024b. Parallelizing linear transformers with the delta rule over sequence length. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4791–4800. Association for Computational Linguistics.

Shuangfei Zhai, Walter Talbott, Nitish Srivastava, Chen Huang, Hanlin Goh, Ruixiang Zhang, and Josh M. Susskind. 2021. An attention free transformer. *CoRR*, abs/2105.14103.

Michael Zhang, Kush Bhatia, Hermann Kumbong, and Christopher Ré. 2024. The hedgehog & the porcupine: Expressive linear attentions with softmax mimicry. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Liang Zhao, Xiachong Feng, Xiaocheng Feng, Weihong Zhong, Dongliang Xu, Qing Yang, Hongtao Liu, Bing Qin, and Ting Liu. 2024. Length extrapolation of transformers: A survey from the perspective of positional encoding. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 9959–9977. Association for Computational Linguistics.

Lin Zheng, Chong Wang, and Lingpeng Kong. 2022. Linear complexity randomized self-attention mechanism. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 27011–27041. PMLR.

A Proof of Theorem 3.1

First, we recall the formula for computing the variance of product and sum of independent random variables. Assume that x and y are independent, then:

$$\text{Var}(xy) = \text{Var}(x)\text{Var}(y) + \text{Var}(x)(E(y))^2 + \text{Var}(y)(E(x))^2$$

$$\text{Var}(x + y) = \text{Var}(x) + \text{Var}(y).$$

Because, $\text{Var}(x_i) = 1$ and $E(x_i) = 0$ (and same for y_i), applying the formulas immediately gives the variance for u.

Next, if $x_i \sim \mathcal{N}(0, 1)$ one can compute the mean of $\exp(x_i)$ and its variance by taking the corresponding integrals (or using the formulas for log-normal distribution). The result will be: $E(\exp(x_i)) = e^{\frac{1}{2}}$, and $\text{Var}(\exp(x_i)) = e(e - 1)$.

Substituting these results to the first formula for the variance of the product, we have:

$$\text{Var}(\exp(x_i))\text{Var}(\exp(y_i)) = e^2(e - 1)^2 + 2e(e - 1)e.$$

Simplifying the expression, we get $e^2(e^2 - 1)$.

The final formula for z follows form the independence of each summand and the formula for the sum of the variance.

A.1 Elaboration on Theorem 3.1

The feature map that we use in our linear transformer is not just the exponential map but the normalized exponential map $\exp(\boldsymbol{x} - \max_i(\boldsymbol{x}_i))$, so the assumption of Theorem 3.1 should be slightly adjusted to be applicable. Let us discuss this in detail.

First, consider independent random variables $x_i \sim \mathcal{N}(0, 1)$, for $i \in [1, d]$. If we subtract $\max_i(x_i)$ from each x_i and consider new variables: $x'_i = x_i - \max_i(x_i)$, they stop being independent and their distribution becomes hard to analyze. Let us simplify the setting to facilitate the analysis. Here $\bar{x} = \max_i(x_i)$ is a random variable by itself, but let us replace it with its expectation. If we assume that d is large, then we can use the following asymptotic (David and Nagaraja, 2005, Example 10.5.3):

$$E(\bar{x}) \approx \sqrt{2 \ln d} + o(1).$$

Now for the simplified analysis of the variance of the feature map, let us subtract not the maximum in the sample $\bar{x}$, but its asymptotic $\sqrt{2 \ln d}$, which is constant and does not depend on the sample. Then we have independent random variables with a new mean. We can reformulate the theorem now:

Theorem A.1 *Consider independent random variables $x_i, y_i \sim \mathcal{N}(-\sqrt{2 \ln d}, 1)$, for $i \in [1, d]$. Define a new variable z by:*

$$z = \sum_{i=1}^{d} \exp(x_i) \times \exp(y_i). \tag{19}$$

Then the variance of z is $e^{-4\sqrt{2 \ln d}} e^2(e^2 - 1)d$ respectively.

The proof is analogous to the previous proof with the following modifications:

$$E(\exp(x_i)) = e^{\frac{1}{2}} e^{-\sqrt{2 \ln d}}$$

and

$$\text{Var}(\exp(x_i)) = e(e - 1)e^{-2\sqrt{2 \ln d}}.$$

Note that $e^{-4\sqrt{2 \ln d}} < 1$, so the previous normalization constant from Theorem 3.1 is the upper bound.

B Detailed Experiment Settings

In this section, we provide the detailed experiment settings for both our training from scratch and post-linearization and continual pre-training experiments. For the transformer model, we train a 12-layer decoder-only network with 12 heads, head dimension = 64, hidden dimension 768, and MLP dimension 3072 for 50,000 steps, which follows the default architectural details of Pythia-160m but with sequential connection and full RoPE embedding layer. All our linear attention models follow the same setting like MLP layer and attention head numbers. For a fair comparison, all our linear attention models use feature dimension 64. For experiments of common sense reasoning, we download the Pythia-160m checkpoint from Huggingface, then replace the softmax attention module with various linear attention modules and perform continual pertaining on the SlimPajama dataset. For optimization, we use the AdamW optimizer with a learning rate 2e-4 and weight decay 0.01. We use batch size 8 and dropout 0.1.