Composite Attention: A Framework for Combining Sequence Mixing Primitives

Harry Jake Cunningham*
Centre for Artificial Intelligence
University College London

Marc Peter Deisenroth
Centre for Artificial Intelligence
University College London

Abstract

Hybrid attention architectures have shown promising success in both equipping self attention with inductive bias for long-sequence modelling and reducing the computational burden of transformers without sacrificing quality. This paper introduces Composite Attention, a theoretical framework for analyzing the combination of sequence mixing primitives in modern deep learning architectures. Utilizing the definition of sequence mixers as structured linear maps, we formalize the composition of sequence mixing primitives as either *sequential* or *recurrent* composition.

1 Introduction

The design space of sequence models has recently exploded due to the successful introduction of long convolutions [17, 3, 8, 26], state-space models [12, 10, 11, 27], linear attention mechanisms [9, 4, 28, 15], and gating [24, 13]. Hybrid attention models [21, 22, 2] seek to make use of these advancement by combining different sequence mixing primitives to improve efficiency and performance. In this work, we examine the combination of sequence mixing primitives within hybrid attention architectures and propose a framework for enhancing the understanding of composite sequence models through analysis of their matrix structures. Specifically, we aim to: 1) Formalize the composition of sequence mixing primitives, 2) Identify gating as a method for diversifying single-head attention, and 3) Recognize convolution-equipped attention mechanisms as a means of encoding local context.

2 Background

Sequence Mixers Modern deep learning sequence mixers rely on the aggregation of information as the weighted sum of tokens. Following [14], we can define any sequence mixer which performs aggregation as a linear map $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, where $\boldsymbol{A} \in \mathbb{R}^{L\times L}$ is a matrix of *attention* weights and $\boldsymbol{x} \in \mathbb{R}^{L\times L}$ is an input sequence of length L and dimension d.

Definition 2.1 (Token Mixing as Linear Maps.)**.** *Let $\boldsymbol{x} \in \mathbb{R}^{L\times D}$ be a sequence of tokens, of length L and embedding dimension D. Let $\mathcal{M} \subseteq \mathbb{R}^{L\times L}$ be a space of possible attention matrices. Let H be the number of heads and P the head dimension such that $HP = D$. For each head $h \in H$, we define a generating function $f^h_{\mathcal{M}} : \mathbb{R}^{L\times D} \times \Theta \to \mathcal{M}$, parameterized by Θ, that maps the input to a space of possible attention matrices. Denoting the attention weight matrix $\boldsymbol{A}^h = f^h_{\mathcal{M}}(\boldsymbol{x}, \Theta) \in \mathbb{R}^{L\times L}$, we define sequence mixing via aggregation as,*

$$\boldsymbol{y}^h = \boldsymbol{A}^h \boldsymbol{x}^h \tag{1}$$

where $\boldsymbol{x}^h \in \mathbb{R}^{L\times P}$ is the input for a given head h, $\boldsymbol{y}^h \in \mathbb{R}^{L\times P}$ is the output for a given head h, and $\boldsymbol{y} = \text{Concat}[\boldsymbol{y}^0, \cdots, \boldsymbol{y}^{h-1}] \in \mathbb{R}^{L\times D}$ is the same shape as the input.

*Correspondence to jake.cunningham.21@ucl.ac.uk

4th NeurIPS Efficient Natural Language and Speech Processing Workshop (ENLSP-IV 2024).

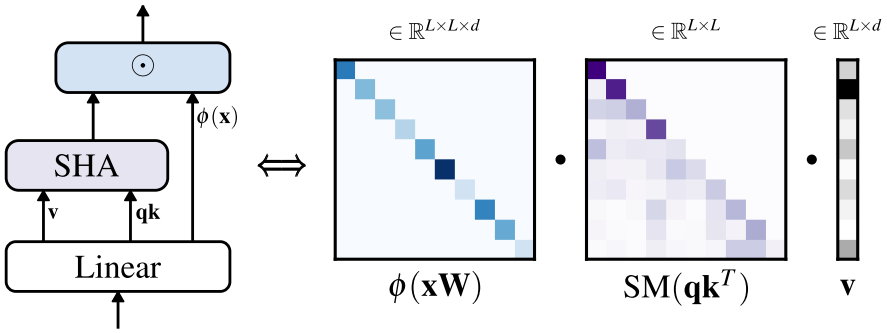

Figure 1: **Gated Attention Unit as Sequential Composition of Sequence Mixers.** (Left) A gated attention unit which consists of single-head attention (SHA) followed by an element-wise product $\odot$, where ϕ is an element-wise activation function. (Right) The equivalent compositional form, denoting a *sequential* input dependency structure and their corresponding generating functions.

For notational ease, we define $f_{\mathcal{M}}(\cdot) = \text{Concat}[f^0_{\mathcal{M}}(\cdot), \cdots, f^{h-1}_{\mathcal{M}}(\cdot)]$ as the generating function that returns an attention matrix $\boldsymbol{A} \in \mathbb{R}^{H\times L\times L}$ for all heads H.

Hybrid Architectures There is a growing interest in designing *compositional* or hybrid architectures that integrate different sequence mixing primitives to obtain a best-of-all-world solution. Gating has proven a powerful method of enhancing a models input dependency [19, 13, 1, 7, 22]. Depthwise separable convolutions placed before linear attention mechanisms have been shown to reduce the quality gap to SoftMax attention [21, 2, 5, 9, 7, 20, 22]. Hybrid architectures which alternate between SoftMax attention layers and simpler sequence mixers have proven to be more computationally efficient and as performant [17, 2, 23, 25, 18].

3 Framework for Composition of Sequence Mixing Primitives

We introduce *Composite Attention*, a framework for analysing the combination of sequence mixing primitives. We can combine input-dependent transformations in 2 different ways: via 1) the *sequential* composition of linear maps where each map depends on the same input $\boldsymbol{x}$ and 2) the *recurrent* composition of linear maps where each map depends on the output of the previous transformation.

Definition 3.1 (Sequential Composition of Sequence Mixers)**.** *Let $\boldsymbol{x} \in \mathbb{R}^{L\times d}$ be a sequence of tokens, of length L and embedding dimension d. Let $\{\mathcal{M}^{(i)}\}_{i=0}^{N-1}$ be a set of classes of attention matrices, where $\mathcal{M}^{(i)} \subseteq \mathbb{R}^{L\times L}$. Let $\{f^{(i)}_{\mathcal{M}}\}_{i=0}^{N-1}$ be a set of generating functions, where each $f^{(i)}_{\mathcal{M}} : \boldsymbol{X} \times \Phi^{(i)} \to \mathcal{M}^{(i)}$ is a function that maps from the input to the space of structured matrices $\mathcal{M}^{(i)}$. We define the sequential composition of Attention matrices as*

$$\boldsymbol{y} = g\left(f^{(N-1)}_{\mathcal{M}}(\boldsymbol{x}), \cdots, f^{(1)}_{\mathcal{M}}(\boldsymbol{x}), f^{(0)}_{\mathcal{M}}(\boldsymbol{x})\right) \cdot \boldsymbol{x} \tag{2}$$

$$= \boldsymbol{A}_S \cdot \boldsymbol{x} \tag{3}$$

where $g : \mathcal{M}^{(0)} \times \mathcal{M}^{(1)} \times \cdots \times \mathcal{M}^{(N-1)} \to \mathbb{R}^{L\times L}$ is a function that takes as input a set of N structured matrices and returns $\boldsymbol{A}_S \in \mathbb{R}^{L\times L}$, denoted the compositional attention matrix corresponding to the composition of attention matrices, which each depend on the input $\boldsymbol{x}$, by the function $g(\cdot)$.

Remark. Typically we assume that $g(\cdot)$ is a composition of binary operations such as matrix additions or matrix multiplications. For the case of matrix multiplications the sequential composition can be constructed as

$$\boldsymbol{y} = \left(f^{(N-1)}_{\mathcal{M}}(\boldsymbol{x}) \cdot \ldots \cdot f^{(1)}_{\mathcal{M}}(\boldsymbol{x}) \cdot f^{(0)}_{\mathcal{M}}(\boldsymbol{x})\right) \cdot \boldsymbol{x} \tag{4}$$

$$= \boldsymbol{A}^{(N-1)} \cdot \ldots \cdot \boldsymbol{A}^{(1)} \cdot \boldsymbol{A}^{(0)} \cdot \boldsymbol{x} \tag{5}$$

where $\boldsymbol{A}_S = \boldsymbol{A}^{(N-1)} \cdot \ldots \cdot \boldsymbol{A}^{(1)} \cdot \boldsymbol{A}^{(0)}$ is the compositional attention matrix. We stress though that our definition is general and encompasses potentially non-linear functions of the input matrices.

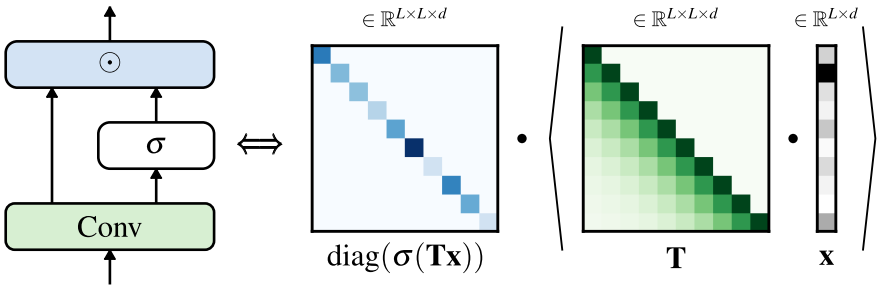

Figure 2: **Silu Activation as Recurrent Composition of Sequence Mixers.** (Left) A global depthwise separable convolution followed by a silu activation function, where σ is the logistic sigmoid applied element-wise. (Right) The equivalent compositional form, denoting a *recurrent* input dependency structure and their corresponding generating functions.

Hierarchical gating, as used in gated attention units (GAU) (See Figure 1), is an example of sequential composition whereby each sequence mixer depends upon the same input.

Definition 3.2 (Recurrent Composition of Sequence Mixers)**.** *Let $\boldsymbol{x} \in \mathbb{R}^{L\times d}$ be a sequence of tokens, of length L and embedding dimension d. Let $\{\mathcal{M}^{(i)}\}_{i=0}^{N-1}$ be a set of classes of attention matrices, where $\mathcal{M}^{(i)} \subseteq \mathbb{R}^{L\times L}$. Let $\{f_{\mathcal{M}}^{(i)}\}_{i=0}^{N-1}$ be a set of generating functions, where each $f_{\mathcal{M}}^{(i)} : \boldsymbol{X} \times \Phi^{(i)} \to \mathcal{M}^{(i)}$ is a function that maps from the input to the space of structure matrices $\mathcal{M}^{(i)}$. We define the sequential composition of Attention matrices as*

$$\boldsymbol{y}_n = f_{\mathcal{M}}^{(i)}(\boldsymbol{y}_{n-1}) \cdot \boldsymbol{y}_{n-1} \tag{6}$$
$$\boldsymbol{y}_0 = \boldsymbol{x} \tag{7}$$

where the output $\boldsymbol{y}_n$ is sequentially generated.

Recurrent composition is typically found in branches of a gated architecture such as a short convolution followed by a silu activation function [6] as used in Mamba [9] (see Figure 2).

4 Composite Attention

We will use our framework to analyze combinations of sequence mixing primitives, focusing on the properties of their attention matrices, generating functions, and matrix products, especially looking at gating and convolution-equipped attention.

4.1 Exploration of Composite Sequence Mixer Framework

We begin with a Gated Attention Unit (GAU), defined as,

$$\boldsymbol{u} = \phi_u(\boldsymbol{x}\boldsymbol{W}_u), \quad \boldsymbol{v} = \phi_v(\boldsymbol{x}\boldsymbol{W}_v), \quad \boldsymbol{g} = \phi_g(\boldsymbol{x}\boldsymbol{W}_g) \tag{8}$$
$$\boldsymbol{q} = \kappa_q \odot \boldsymbol{u} + \mu_q, \quad \boldsymbol{k} = \kappa_k \odot \boldsymbol{u} + \mu_k \tag{9}$$
$$\boldsymbol{a} = \text{SoftMax}\left(\frac{\boldsymbol{q}\boldsymbol{k}^T}{\sqrt{d}}\right)\boldsymbol{v} \tag{10}$$
$$\boldsymbol{y} = \boldsymbol{a} \odot \boldsymbol{g} \tag{11}$$

This structure can be described similarly to hierarchical gating as the *sequential composition* of attention matrices, where both *self-attention* and *gating* correspond to input-dependent sequence mixers whose generating function depends on the input sequence $\boldsymbol{x}$ as,

$$\boldsymbol{y} = \text{diag}(\phi_g(\boldsymbol{x}\boldsymbol{W}_g)) \cdot \text{SoftMax}\left(\frac{\boldsymbol{q}\boldsymbol{k}^T}{\sqrt{d}}\right) \cdot \phi_u(\boldsymbol{x}\boldsymbol{W}_v) \tag{12}$$
$$= f_{\boldsymbol{G}}(\boldsymbol{x}) \cdot f_{\boldsymbol{A}}(\boldsymbol{x}) \cdot \phi_u(\boldsymbol{x}\boldsymbol{W}_v) \tag{13}$$
$$= \boldsymbol{A}_G \cdot \boldsymbol{A}_A \cdot \boldsymbol{v} \tag{14}$$

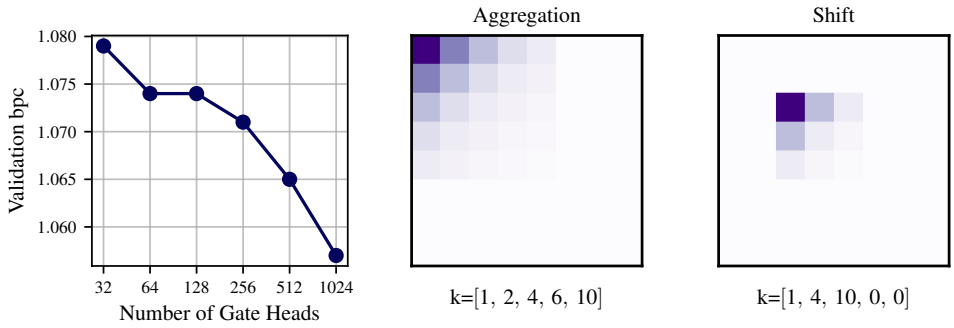

Figure 3: **Left.** Gating Heads Ablation on enwik8. Performance of *GAU* improves with increasing head dimension on `enwik8` language modelling task. **Center.** Outer product of unit impulse convolved with an *aggregation* filter. Shows how convolution equipped attention mechanisms spread token information across the attention matrix, encoding context. **Right**. Outer product of unit impulse convolved with a *shift* filter. Demonstrates ability of local convolutions to perform local token shifts, acting as a form of memory.

Gating Diversifies Single-Head Attention. As a sequence mixer gating can be represented as an $L \times L$ diagonal matrix, with generating function $f_{\boldsymbol{G}}(\boldsymbol{x}) = \text{diag}(\phi(\boldsymbol{x}\boldsymbol{W})^T) \in \mathbb{R}^{L\times L\times h_g}$ where h_g corresponds to the number of gating heads, which is typically set to the hidden dimension d. We *hypothesize* that the effectiveness of gating is derived from its use of a very large number of heads which when composed with single-head attention (SHA) or few-head attention diversifies the attention layer into a multi-headed linear map over the sequence dimension

$$\boldsymbol{A}_G = f_G(\boldsymbol{x}) = \text{diag}(\phi(\boldsymbol{x}\boldsymbol{W})^T) \qquad \in \mathbb{R}^{L\times L\times h_g} \tag{15}$$

$$\boldsymbol{A}_{SHA} = f_{SHA}(\boldsymbol{x}) = \text{SoftMax}\left(\frac{\boldsymbol{q}\boldsymbol{k}^T}{\sqrt{d}}\right) \qquad \in \mathbb{R}^{L\times L\times 1} \tag{16}$$

$$\boldsymbol{A}_{GAU} = \boldsymbol{A}_G \cdot \text{Repeat}(\boldsymbol{A}_{SHA}, h_g) \qquad \in \mathbb{R}^{L\times L\times h_g} \tag{17}$$

We emphasize the importance of using many heads when gating by conducting an Ablation study using the MEGA transformer [21]. Indeed, we find that increasing the number of gating heads improves performance on the `enwik8` dataset (see Figure 3)

Convolutions Encode Context. Placing convolutional layers prior to computing the $\boldsymbol{q}, \boldsymbol{k}$ projections has been proposed as an alternative to positional encodings [16, 21] as well as a means of introducing locality which is required to capture long-range dependencies by means of a hierarchical combination of local dependencies [4, 7, 3]. We examine the effect of convolution equipped attention by *composing together* the convolution operator with the generating function for self-attention.

Definition 4.1 (Convolution Equipped Attention)**.** *Let $\boldsymbol{x} \in \mathbb{R}^{L\times d}$ be an input of length L and dimension d, $\boldsymbol{k}_q = [\boldsymbol{\alpha}_0, \cdots, \boldsymbol{\alpha}_{L-1}] \in \mathbb{R}^{L\times d}$ and $\boldsymbol{k}_k = [\boldsymbol{\beta}_0, \cdots, \boldsymbol{\beta}_{L-1}] \in \mathbb{R}^{L\times d}$ the query and key convolution kernels of length L and $\boldsymbol{W}_q, \boldsymbol{W}_k \in \mathbb{R}^{d\times d}$ dense linear projections over the channel dimension d. Let the query and keys be defined as $\boldsymbol{q} = (\boldsymbol{k}_q * \boldsymbol{x})\boldsymbol{W}_q$ and $\boldsymbol{k} = (\boldsymbol{k}_k * \boldsymbol{x})\boldsymbol{W}_k$ respectively. Hence,*

$$\text{SoftMax}\left(\frac{\boldsymbol{q}_i^T\boldsymbol{k}_j}{\sqrt{d}}\right) = \text{SoftMax}\left(\frac{1}{\sqrt{d}}\sum_{n=0}^{i}\sum_{m=0}^{j}(\boldsymbol{\alpha}_{i-n}\odot\boldsymbol{x}_n)^T\boldsymbol{W}_q^T\boldsymbol{W}_k(\boldsymbol{\beta}_{j-m}\odot\boldsymbol{x}_m)\right) \tag{18}$$

corresponds to the convolution equipped attention matrix.

Here we can see how convolutions are able to spread the influence of tokens across the attention matrix through the convolution weights α and β. We note that convolutions can do this in 2 ways: 1) by aggregating local tokens providing positional information and 2) by performing local token shifts (e.g. $[0, 1, 1] * [a, b, c] = [0, a, b]$) acting as a form of memory [7, 12]. We illustrate both of these effects in Figure 3 by convolving a unit impulse with convolution kernels corresponding to *aggregations* and *shifts* and plotting the outer product $\boldsymbol{q}\boldsymbol{k}^T$.

Further, we can see how the dot product between tokens in standard SoftMax attention has been replaced by a sum of dot products that evaluate the similarity between tokens that appear before the tokens of interest. Hence, convolution equipped attention amplifies attention weights for tokens whose contexts are similar.

We define *convolution equipped GAU* by the following composition,

$$\boldsymbol{y} = f_{\boldsymbol{G}}(f_{\boldsymbol{C}_1}(\boldsymbol{x})) \cdot f_{\boldsymbol{A}}(f_{\boldsymbol{C}_0}(\boldsymbol{x})) \cdot \boldsymbol{v} \tag{19}$$

$$= f_{\boldsymbol{G}}(\boldsymbol{T}_1 \cdot \boldsymbol{x})) \cdot f_{\boldsymbol{A}}(\boldsymbol{T}_0 \cdot \boldsymbol{x})) \cdot \boldsymbol{v} \tag{20}$$

where $\mathbf{T}_0$ and $\mathbf{T}_1$ correspond to Toeplitz matrices representing the non-input dependent convolutions, which are followed by a linear projection and silu activation defined within the generating function $f_{\boldsymbol{A}}$, corresponding to a *recurrent* composition.

5 Conclusion

We have developed Composite Attention, a framework for combining sequence mixing primitives using both sequential and recurrent compositions of matrix structures. Through our framework, we analyze the impact of gating and convolutions in a gated attention unit. We find that gating enhances single-head attention by incorporating multiple heads, while convolutions encode context by performing local aggregation and shifting. Looking ahead, we aim to utilize our framework to create new sub-quadratic architectures, with a specific focus on improving linear attention mechanisms using cost-effective sequence mixers.

References

[1] S. Arora, S. Eyuboglu, A. Timalsina, I. Johnson, M. Poli, J. Zou, A. Rudra, and C. Ré. Zoology: Measuring and improving recall in efficient language models. *arXiv preprint arXiv:2312.04927*, 2023.

[2] S. Arora, S. Eyuboglu, M. Zhang, A. Timalsina, S. Alberti, D. Zinsley, J. Zou, A. Rudra, and C. Ré. Simple linear attention language models balance the recall-throughput tradeoff. *arXiv preprint arXiv:2402.18668*, 2024.

[3] H. J. Cunningham, G. Giannone, M. Zhang, and M. P. Deisenroth. Reparameterized multi-resolution convolutions for long sequence modelling. *arXiv preprint arXiv:2408.09453*, 2024.

[4] T. Dao and A. Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

[5] S. De, S. L. Smith, A. Fernando, A. Botev, G. Cristian-Muraru, A. Gu, R. Haroun, L. Berrada, Y. Chen, S. Srinivasan, et al. Griffin: Mixing gated linear recurrences with local attention for efficient language models. *arXiv preprint arXiv:2402.19427*, 2024.

[6] S. Elfwing, E. Uchibe, and K. Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018.

[7] D. Y. Fu, T. Dao, K. K. Saab, A. W. Thomas, A. Rudra, and C. Ré. Hungry hungry hippos: Towards language modeling with state space models. *arXiv preprint arXiv:2212.14052*, 2022.

[8] D. Y. Fu, E. L. Epstein, E. Nguyen, A. W. Thomas, M. Zhang, T. Dao, A. Rudra, and C. Ré. Simple hardware-efficient long convolutions for sequence modeling. In *International Conference on Machine Learning*, pages 10373–10391. PMLR, 2023.

[9] A. Gu and T. Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[10] A. Gu, K. Goel, and C. Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

[11] A. Gu, K. Goel, A. Gupta, and C. Ré. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems*, 35:35971–35983, 2022.

[12] A. Gu, I. Johnson, A. Timalsina, A. Rudra, and C. Ré. How to train your hippo: State space models with generalized orthogonal basis projections. *arXiv preprint arXiv:2206.12037*, 2022.

[13] W. Hua, Z. Dai, H. Liu, and Q. Le. Transformer quality in linear time. In *International conference on machine learning*, pages 9099–9117. PMLR, 2022.

[14] S. Hwang, A. Lahoti, T. Dao, and A. Gu. Hydra: Bidirectional state space models through generalized matrix mixers. *arXiv preprint arXiv:2407.09941*, 2024.

[15] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

[16] M. Li, X. Zhang, Y. Huang, and S. Oymak. On the power of convolution augmented transformer. *arXiv preprint arXiv:2407.05591*, 2024.

[17] Y. Li, T. Cai, Y. Zhang, D. Chen, and D. Dey. What makes convolutional models great on long sequence modeling? *arXiv preprint arXiv:2210.09298*, 2022.

[18] O. Lieber, B. Lenz, H. Bata, G. Cohen, J. Osin, I. Dalmedigos, E. Safahi, S. Meirom, Y. Belinkov, S. Shalev-Shwartz, et al. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887*, 2024.

[19] H. Liu, Z. Dai, D. So, and Q. V. Le. Pay attention to mlps. *Advances in neural information processing systems*, 34:9204–9215, 2021.

[20] Z. Liu, S. Li, L. Wang, Z. Wang, Y. Liu, and S. Z. Li. Short-long convolutions help hardware-efficient linear attention to focus on long sequences. *arXiv preprint arXiv:2406.08128*, 2024.

[21] X. Ma, C. Zhou, X. Kong, J. He, L. Gui, G. Neubig, J. May, and L. Zettlemoyer. Mega: moving average equipped gated attention. *arXiv preprint arXiv:2209.10655*, 2022.

[22] X. Ma, X. Yang, W. Xiong, B. Chen, L. Yu, H. Zhang, J. May, L. Zettlemoyer, O. Levy, and C. Zhou. Megalodon: Efficient llm pretraining and inference with unlimited context length. *arXiv preprint arXiv:2404.08801*, 2024.

[23] J. Park, J. Park, Z. Xiong, N. Lee, J. Cho, S. Oymak, K. Lee, and D. Papailiopoulos. Can mamba learn how to learn? a comparative study on in-context learning tasks. *arXiv preprint arXiv:2402.04248*, 2024.

[24] M. Poli, S. Massaroli, E. Nguyen, D. Y. Fu, T. Dao, S. Baccus, Y. Bengio, S. Ermon, and C. Ré. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*, pages 28043–28078. PMLR, 2023.

[25] L. Ren, Y. Liu, Y. Lu, Y. Shen, C. Liang, and W. Chen. Samba: Simple hybrid state space models for efficient unlimited context language modeling. *arXiv preprint arXiv:2406.07522*, 2024.

[26] D. W. Romero, A. Kuzina, E. J. Bekkers, J. M. Tomczak, and M. Hoogendoorn. Ckconv: Continuous kernel convolution for sequential data. *arXiv preprint arXiv:2102.02611*, 2021.

[27] J. T. Smith, A. Warrington, and S. W. Linderman. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*, 2022.

[28] S. Yang, B. Wang, Y. Shen, R. Panda, and Y. Kim. Gated linear attention transformers with hardware-efficient training. *arXiv preprint arXiv:2312.06635*, 2023.

A Appendix

A.1 Convolutions Encode Context

Let $\boldsymbol{x} \in \mathbb{R}^{L \times d}$ be an input. We define the queries and key vectors as,

$$\boldsymbol{q} = (\boldsymbol{k}_q * \boldsymbol{x})\boldsymbol{W}_q = \boldsymbol{T}_q \boldsymbol{x} \boldsymbol{W}_q \tag{21}$$

$$\boldsymbol{k} = (\boldsymbol{k}_k * \boldsymbol{x})\boldsymbol{W}_k = \boldsymbol{T}_k \boldsymbol{x} \boldsymbol{W}_k \tag{22}$$

where $\boldsymbol{T}_q, \boldsymbol{T}_k \in \mathbb{R}^{L \times L \times d}$ are Toeplitz matrices which represent a depthwise separable convolution as a matrix-vector product over the sequence length dimension L and $\boldsymbol{W}_q, \boldsymbol{W}_k \in \mathbb{R}^{d \times d}$ are dense linear projections over the channel dimension d.

Splitting the convolution matrix and input into d heads we compute the depthwise separable convolution for a single head i as

$$\boldsymbol{T}_q^i \boldsymbol{x}^i = \begin{bmatrix} \alpha_0^i & 0 & 0 & 0 & 0 & 0 \\ \alpha_1^i & \alpha_0^i & 0 & 0 & 0 & 0 \\ \vdots & \alpha_1^i & \alpha_0^i & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \alpha_{L-2}^i & \ddots & \ddots & \alpha_1 & \alpha_0^i & 0 \\ \alpha_{L-1}^i & \alpha_{L-2}^i & \dots & \dots & \alpha_1^i & \alpha_0^i \end{bmatrix} \begin{bmatrix} x_0^i \\ x_1^i \\ \vdots \\ \vdots \\ \vdots \\ x_{L-1}^i \end{bmatrix} = \begin{bmatrix} \alpha_o^i x_0^i \\ \alpha_1^i x_0^i + \alpha_0^i x_1^i \\ \vdots \\ \vdots \\ \vdots \\ \sum_{l=0}^{L-1} \alpha_{L-1-l}^i x_l^i \end{bmatrix} \in \mathbb{R}^L \tag{23}$$

and hence for all heads,

$$\boldsymbol{T}_q \boldsymbol{x} = \begin{bmatrix} \boldsymbol{\alpha_o} \odot \boldsymbol{x}_0 \\ \boldsymbol{\alpha_1} \odot \boldsymbol{x}_0 + \boldsymbol{\alpha}_0 \boldsymbol{x}_1 \\ \vdots \\ \vdots \\ \vdots \\ \sum_{l=0}^{L-1} \boldsymbol{\alpha}_{L-1-l} \odot \boldsymbol{x}_l \end{bmatrix} \in \mathbb{R}^{L \times d} \tag{24}$$

such that the queries and keys are represented as,

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{W}_q(\boldsymbol{\alpha_o} \odot \boldsymbol{x}_0) \\ \boldsymbol{W}_q(\boldsymbol{\alpha_1} \odot \boldsymbol{x}_0 + \boldsymbol{\alpha}_0 \odot \boldsymbol{x}_1) \\ \vdots \\ \vdots \\ \vdots \\ \sum_{l=0}^{L-1} \boldsymbol{W}_q(\boldsymbol{\alpha}_{L-1-l} \odot \boldsymbol{x}_l) \end{bmatrix} \in \mathbb{R}^{L \times d}, \quad \boldsymbol{k} = \begin{bmatrix} \boldsymbol{W}_k(\boldsymbol{\beta_o} \odot \boldsymbol{x}_0) \\ \boldsymbol{W}_k(\boldsymbol{\beta_1} \odot \boldsymbol{x}_0 + \boldsymbol{\beta}_0 \odot \boldsymbol{x}_1) \\ \vdots \\ \vdots \\ \vdots \\ \sum_{l=0}^{L-1} \boldsymbol{W}_k(\boldsymbol{\beta}_{L-1-l} \odot \boldsymbol{x}_l) \end{bmatrix} \in \mathbb{R}^{L \times d} \tag{25}$$

Considering the case of single-head attention, the outer product $\boldsymbol{q}\boldsymbol{k}^T$ is defined as,

$$\boldsymbol{q}_i^T \boldsymbol{k}_j = \sum_{n=0}^{i} \sum_{m=0}^{j} (\boldsymbol{\alpha}_{i-n} \odot \boldsymbol{x}_n)^T \boldsymbol{W}_q^T \boldsymbol{W}_k (\boldsymbol{\beta}_{j-m} \odot \boldsymbol{x}_m) \tag{26}$$