Short-Long Convolutions Help Hardware-Efficient Linear Attention to Focus on Long Sequences

Zicheng Liu [1] **Siyuan Li** [1] **Li Wang** [1] **Zedong Wang** [1] **Yunfan Liu** [1] **Stan Z. Li**[† 1]

Abstract

To mitigate the computational complexity in the self-attention mechanism on long sequences, linear attention utilizes computation tricks to achieve linear complexity, while state space models (SSMs) popularize a favourable practice of using non-data-dependent memory pattern, *i.e.,* emphasize the near and neglect the distant, to processing sequences. Recent studies have shown the priorities by combining them as one. However, the efficiency of linear attention remains only at the theoretical level in a causal setting, and SSMs require various designed constraints to operate effectively on specific data. Therefore, in order to unveil the true power of the hybrid design, the following two issues need to be addressed: (1) hardware-efficient implementation for linear attention and (2) stabilization of SSMs. To achieve this, we leverage the thought of tiling and hierarchy to propose CHELA (short-long **C**onvolutions with **H**ardware-**E**fficient **L**inear **A**ttention), which replaces SSMs with short-long convolutions and implements linear attention in a divide-and-conquer manner. This approach enjoys global abstraction and data-dependent selection from stable SSM and linear attention while maintaining real linear complexity. Our comprehensive experiments on the Long Range Arena benchmark and language modeling tasks demonstrate the effectiveness of the proposed method.

1. Introduction

Transformer models have demonstrated remarkable performance on a range of natural language processing tasks (Vaswani et al., 2017), such as language modeling (Devlin et al., 2019), visual signal processing (Dosovitskiy et al., 2021; Liu et al., 2022; Li et al., 2023; Liu et al., 2023), and speech understanding (Gulati et al., 2020). These models use the attention mechanism, which calculates a dependency score for each pair of tokens in an input sequence. Consequently, full attention has a quadratic time and space complexity relative to the sequence length. This complexity, however, becomes computationally prohibitive for tasks that involve long sequences (Lin et al., 2022). It is worth mentioning that Transformer models equipped with full attention tend to overfit. This is because the attention mechanism does not make any assumptions about the structure of the inputs, which leads to the absence of structural biases. To train a Transformer model, even the order information has to be included. Therefore, the full attention is too flexible to overfit to noise. This limitation restricts the practicality of these models in long sequence modeling, where the dependency signal is often weak and the signal-to-noise ratio is low. To solve this, recent studies have designed hybrid models (Ma et al., 2022; Zuo et al., 2023) by combining efficient state space models (SSMs) (Gu et al., 2021; 2020a; 2022; Hasani et al., 2022; Smith et al., 2023), with expressive attention variants for modeling long sequences from perspectives in structured and flexible patterns, achieving promising results. It is worth noting that their complexity is essentially quadratic, and the corresponding linear versions both suffer performance degradation (see Fig. 1 *left*).

While it is true that the simple use of a chunk linearization strategy can rival the speed of SSMs, the drop in performance is significant. We, therefore, abandon this strategy and the softmax function to accelerate the linear attention operations of the kernel-based approach. In addition, long convolution (Fu et al., 2023b) is more efficient and easier to implement than the traditional SSMs. We believe combining these two modules could release the true power of the hybrid model with faster speed and better performance. However, the existence of such a dilemma is worth considering: **(1) the linear attention in hybrid models is a speed bottleneck for long convolution, the comparison is shown in Fig. 1 *right*; (2) the instability of long convolution makes attention hard to optimize.** Specifically, due to the intense memory access (I/O) on the GPU and cumulative summation (`cumsum`) operation in a casual setting,

[1]AI Lab, Research Center for Industries of the Future, Westlake University, Hangzhou, China. Correspondence to: Stan Z. Li <stan.zq.li@westlake.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

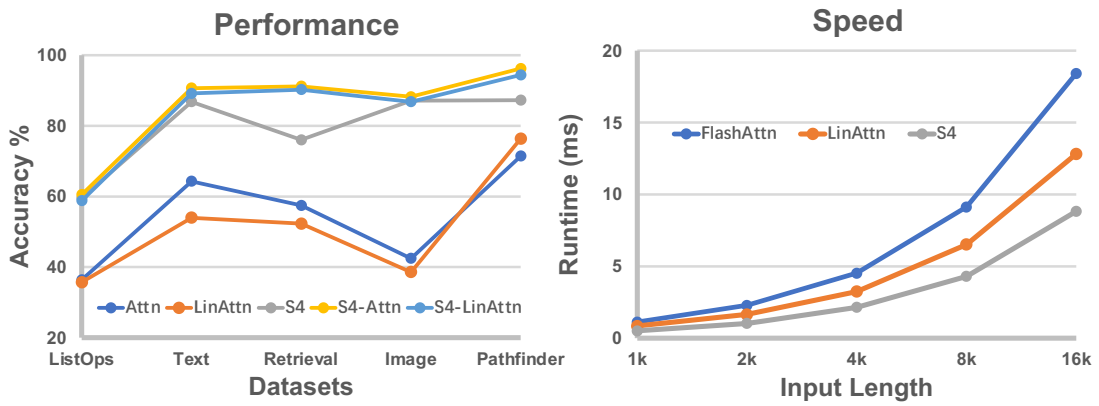

Figure 1. Demonstration of (*left*) comparison of various popular models on the Long Range Arena Dataset, and (*right*) speed benchmark on different implementations of attentions. Figure *left*, despite showing promising performance of SSM-Attention models, the linear version of these models degenerated. Figure *right* benchmarks attention speed with around 200M parameters, showing linear attention is not linear with sequence length and is significantly slower than S4, which can be a speed bottleneck in these hybrid designs.

the notable reduction in complexity from $\mathcal{O}(L^2)$ to $\mathcal{O}(L)$ in linear attention is only theoretical (Qin et al., 2024); On the other hand, the long convolution needs hand-crafted regularizations to prevent over-fitting high-frequency noise.

To escape the dilemma, we have developed a novel model called CHELA, which stands for short-long **C**onvolutional with **H**ardware-**E**fficient **L**inear **A**ttention. This model is designed to effectively and efficiently capture complex dependencies in long inputs. Inspired by FlashAttention (Dao et al., 2022; Dao, 2023), the model comprises multiple layers of real linear token mixers with hardware-friendly implementation with a gating mechanism that achieves comparable performance to full attention. We have added a stabilizer, reparameterizable (Ding et al., 2022; 2023) short-long convolutions, to the bottom layer of the long convolution to integrate multiple frequencies with inputs. As a result, the top layers of CHELA capture more refined data-dependent information using linear attention, while the convolutions add a structural bias that enhances global abstraction in multi-level frequencies. We demonstrate the efficiency and effectiveness of CHELA on various datasets and tasks. First, we show that the proposed method outperforms existing approaches on the Long Range Arena (LRA) benchmark (Tay et al., 2020b), which is designed to test models' ability in modeling long sequences. Second, we show that in autoregressive language modeling, CHELA is not only significantly faster than the vanilla Transformer but also yields better performance. In all the settings, CHELA outperforms the baselines. Finally, we provide further analysis and ablation experiments to demonstrate the effectiveness.

Our contribution can be summarized in three folds:

- **Efficient Implementation**: We release the potential speed of linear attention in the attention-ssm hybrid model, maintaining a constant rate regardless of sequence length with fixed memory.
- **New Design**: We improve multi-frequency learning with multi-level convolutional hierarchies, leading to better performance and stability for long convolution.
- **Promising Results:** By connecting the above two basic modules, we propose CHELA for efficient long sequence learners are capable of both global information abstraction and fine-grained data-dependent selection.

2. Background

2.1. Self Attention

If we have an input $\mathbf{X}$ that belongs to $\mathbb{R}^{L\times d}$, where L represents sequence length and d represents the embedding dimension, the attention mechanism produces pair-wise scores denoted as $\mathbf{M}$:

$$\mathbf{M} = \mathrm{Attn}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \sigma\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V}, \quad \text{where } \mathbf{Q}=\mathbf{X}\mathbf{W}_q,\ \mathbf{K}=\mathbf{X}\mathbf{W}_k,\ \mathbf{V}=\mathbf{X}\mathbf{W}_v. \tag{1}$$

Here $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{d\times d}$ are learnable weights, and σ is the attention function. Denote the attention matrix $\mathbf{M} \in \mathbb{R}^{L\times L}$. $\mathbf{M}_{ij}$ captures the relations between the i-th and the j-th input tokens.

Linear attention Taking Norm as σ (Qin et al., 2024), to take advantage of the computational efficiency inherent in right matrix multiplication, the attention score $\mathbf{M}$ can be transformed into its linear variant, which is mathematically equivalent due to the properties of matrix multiplication.

$$\mathbf{M}_{\texttt{linear}} = \texttt{Norm}(\mathbf{Q}(\mathbf{K}^\top\mathbf{V})) \tag{2}$$

2.2. Linear Recurrent Models

Continuous formulation. A model with a continuous time latent space transforms a one-dimensional input signal $u(t)$ into a latent state $x(t)$ that has d_s dimensions, and then this $x(t)$ is transformed into a one-dimensional output signal $y(t)$. Clearly, we can define the process:

$$x'(t) = \mathbf{A}x(t) + \mathbf{B}u(t), \quad y(t) = \mathbf{C}x(t). \tag{3}$$

where $\mathbf{A} \in \mathbb{R}^{d_s \times d_s}$, $\mathbf{B} \in \mathbb{R}^{d_s}$ and $\mathbf{C} \in \mathbb{R}^{d_s}$.

Eq. 3 is utilized in prior research to model long sequences. For instance, Gu et al. (2020a) suggests a set of matrices called HiPPO (high-order polynomial projection operators) to initialize $\mathbf{A}$. The HiPPO matrices are constructed to enable the state $x(t)$ at time t to remember the history of the input $u(t)$ up to the present.

Discrete formulation. Discrete sequences, like natural language inputs $(u_0, u_1, \cdots, u_L)$ with L representing sequence length, are frequently encountered. The model described in Eq. 3 can be discretized (bilinear method, for example) with a step size of Δ to model discrete data.

$$\begin{aligned} x_k &= \overline{\mathbf{A}}x_{k-1} + \overline{\mathbf{B}}u_k, \quad y_k = \overline{\mathbf{C}}x_k, \\ \text{where } \overline{\mathbf{A}} &= (\mathbf{I} - \Delta/2 \cdot \mathbf{A})^{-1}(\mathbf{I} + \Delta/2 \cdot \mathbf{A}), \\ \overline{\mathbf{B}} &= (\mathbf{I} - \Delta/2 \cdot \mathbf{A})^{-1}\Delta\mathbf{B}, \quad \overline{\mathbf{C}} = \mathbf{C}. \end{aligned} \tag{4}$$

We can expand the recurrent representation above to obtain:

$$y_k = \overline{\mathbf{C}}\overline{\mathbf{A}}^k\overline{\mathbf{B}}u_0 + \cdots + \overline{\mathbf{C}}\overline{\mathbf{A}}\overline{\mathbf{B}}u_{k-1} + \overline{\mathbf{C}}\overline{\mathbf{B}}u_k.$$

Convolution formulation. This can be written as a convolutional representation $y = \overline{\mathbf{K}} * u$, where the convolution kernel

$$\overline{\mathbf{K}} \in \mathbb{R}^L = \left(\overline{\mathbf{C}}\overline{\mathbf{B}}, \overline{\mathbf{C}}\overline{\mathbf{A}}\overline{\mathbf{B}}, \cdots, \overline{\mathbf{C}}\overline{\mathbf{A}}^{L-1}\overline{\mathbf{B}}\right). \tag{5}$$

Efficient calculation of the output y in Eq. 5 can be achieved when the convolution kernel $\overline{K}$ is known. However, determining the kernel is a difficult undertaking, and the majority of existing algorithms necessitate a time and space complexity of $O(L^2)$.

S4 model. Gu et al. (2021) devised the S4 model to effectively calculate Eq. 5. In particular, the initialization of $\mathbf{C}$ in Eq. 3 is random, and both $\mathbf{A}$ and $\mathbf{B}$ are initialized as

$$\begin{aligned} &\mathbf{A} = \mathbf{A}^{(d_s)} - \mathbf{P}\mathbf{P}^\top, \quad \mathbf{B}_i = (2i+1)^{\frac{1}{2}}, \\ &\text{where } \mathbf{P}_i = (i + 1/2)^{1/2}, \\ &\mathbf{A}_{ij}^{(d_s)} = -\begin{cases} (i+\frac{1}{2})^{\frac{1}{2}}(j+\frac{1}{2})^{\frac{1}{2}}, & i > j, \\ \frac{1}{2}, & i = j, \\ -(i+\frac{1}{2})^{\frac{1}{2}}(j+\frac{1}{2})^{\frac{1}{2}}, & i < j. \end{cases} \end{aligned} \tag{6}$$

The convolution kernel $\overline{\mathbf{K}}$ in Eq. 5 can be efficiently computed using $O(L)$ time and space complexity. After that, the efficient computation of $y = \overline{\mathbf{K}} * u$ for an input u can be performed.

Long convolution. If we parameterize the kernel with the same length as the sequence in Eq. 5 (Fu et al., 2023b), then we can replace the SSM layer with a learned convolution kernel as a drop-in replacement. The computation can be efficiently done by using the FFT theorem:

$$y = u * \mathbf{K} = \mathbf{F}_L^{-1}\mathbf{D}_\mathbf{K}\mathbf{F}_L u \tag{7}$$

where $\mathbf{F}_L$ denotes the DFT matrix of size L, and $D_K = \text{diag}(\mathbf{F}_L K)$. This so-called FFT convolution scales in linear complexity when dealing with sequences with length L.

3. Why Combining Long Conv and Attention?

As mentioned in Section 1, both long convolution and linear attention mechanisms have their limitations in spite of their widespread use and impressive accomplishments in sequence modeling. By bridging their benefits together, we push the limitation of the hybrid model, which allows us to benefit from the strong inductive bias of structured memory while still having the ability to learn complex data-dependent patterns in true linear time with respect to length. The following two issues are the two main obstacles to extreme performance in long-sequence modeling.

Linear Attention is the Efficiency Bottleneck. We can see from Fig. 1 *right* that as the input sequence grows, the actual efficiency of linear attention is much lower than that of S4, despite the fact that the complexity of the S4 model is $\mathcal{O}(L \log L)$ as for long convolution and $\mathcal{O}(L)$ for linear attention. The two obstacles to achieving a practical wall-clock speedup are intensive GPU memory access and cumulative summation (`cumsum`) in a casual setting. Inspired by Flashattention (Dao et al., 2022; Dao, 2023), we effectively doubled the speed of linear attention.

Long Convolution Needs Short Convolutions. To match the efficiency of hardware-efficient linear attention, the long convolution is considered as a simpler and faster SSM model for long sequence abstraction. However, it requires additional constraint regularizations for stable performance on different data types. This is because it is hard for a single long convolutional kernel to learn multiple frequencies simultaneously, *i.e.,* token patterns that repeat a lot versus those that occur more sparsely. We thereby introduce multi-resolution short convolutional kernels to share the learning pressure of long convolution. They can also be reparameterized by simple linear transformation into a single kernel to speed up the inference phase.

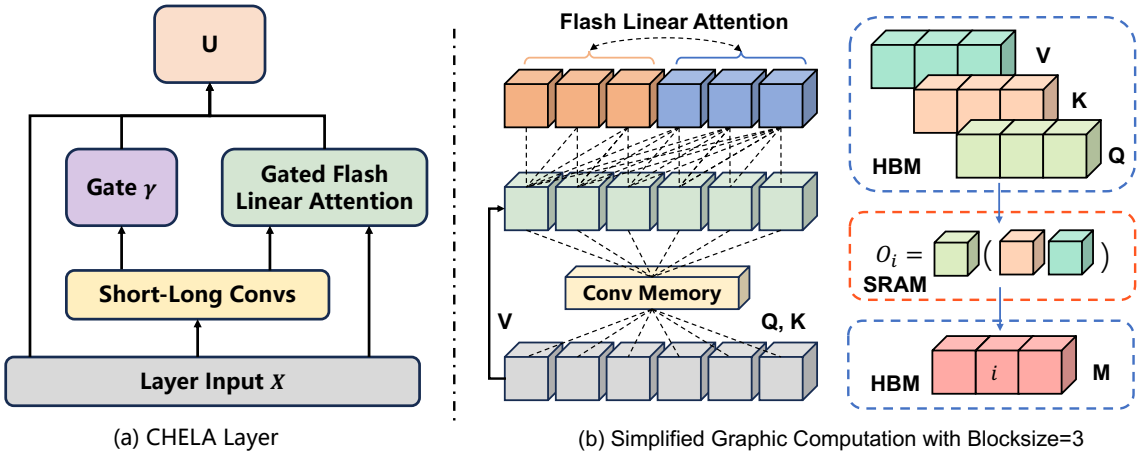

Figure 2. CHELA-model architecture. Figure (a) shows the CHELA layer. Figure (b) is a simplified graphic demonstration of the key components of CHELA: Short-Long Convolutions and Flash Linear Attention. HBM and SRAM denote high bandwidth memory and static random-access memory in GPU, respectively.

4. CHELA: Short-Long Convolutions with Hardware-Efficient Linear Attention

In this section, motivated by the above issues, we propose CHELA, short-long convolutions with hardware-efficient linear attention, which fully enjoys the benefits of each. We first describe how the tiling method is used in linear attention to reach the theoretical performance. We then introduce a simple and effective module named short-long convolutions to enhance the ability of global abstraction in this hybrid design. The blocks of CHELA are demonstrated in detailed architecture, including feed-forward and normalization layers. Moreover, we also discuss the relationship between CHELA and two closely related hybrid models: MEGA (Ma et al., 2022), SPADE (Zuo et al., 2023).

4.1. Hardware-Efficient Linear Attention

In response to the first question, we follow a tiling approach from GLA (Yang et al., 2024), which employs the classic divide-and-conquer strategy to take advantage of the significant difference in memory bandwidth between HBM and SRAM within GPU. Given the inputs $\mathbf{Q}, \mathbf{K}, \mathbf{V}$, we aim to compute the linear attention in SRAM and finally output $\mathbf{O}$. The right-hand multiplication of linear attention is used for the loops between and in the data blocks; noticeably, the left multiplication is used inside the block when an attention mask exists. The intermediate variable $\mathbf{I}$ is iteratively saved and accumulated within SRAM. Subsequently, the outputs of outer and inner loops are summed within SRAM, and the results are returned to HBM. This method aims to capitalize on the distinct advantages of each memory component, optimizing the computational workflow of linear attention to reach its theoretical speed as much as possible. The graphic demonstration is visualized in Fig. 2 (b).

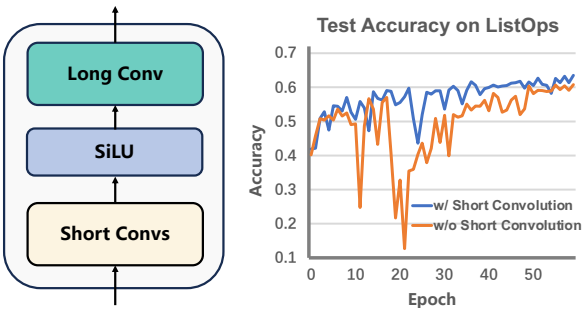

Figure 3. Illustration Short-long Module and the effect of Short Convolution. Figure *left* shows the details of the module structure. Figure *right* proves the role of the short kernel.

4.2. Short-Long Convolutions

CHELA introduces a modification of the long convolution mentioned in Sec. 2.2 by adding short convolutions before the long kernel, named *short-long convolutions*, to learn multi-frequencies information and improve its stability and capacity. Structural Reparameterization (SR) can fuse these short kernels into a single kernel in the inference phase.

Short convolution improves long convolution. The long convolutions have shown success when dealing with long sequences and are much more efficient than conventional SSMs. Empirical results tell us how the weights are initialized and how the regular terms are designed, all of which greatly impact the final performance on specific data types. One important finding is that long convolutions trying to learn both low and high frequencies in the input signal are the central cause of their instability because their kernel mixes various high and low-frequency patterns compared

to S4 (Fu et al., 2023b). Inspired by this, we try adding parallel short kernels in long convolution. As shown in Fig 3, we simply put the short kernels and a SiLU activation function ϕ_{silu} (Ramachandran et al., 2017) before the long convolution. Formally, the computation can be written as:

$$\mathbf{Z} = \overline{\mathbf{K}}_l(\phi_{\text{silu}}(\overline{\mathbf{K}}_s(\mathbf{X}))) \tag{8}$$

where $\overline{\mathbf{K}}_l$ and $\overline{\mathbf{K}}_s$ denote for long and short convolutions. The Fig. 3 *right* also demonstrates the performance of this design on the ListOps dataset: better stabilization and accuracy on the task of long-sequence logic inference. In our experiments, we add two different sizes of short kernels. One is fixed the size at three, and the other varies with the sequence length, whose size equals $2\log_{10} L + 1$.

Structural reparamterization. Structural Reparameterization (Ding et al., 2022; 2023) is a methodology of equivalently converting model structures via linear parameter transformation. For example, RepVGG targeted a deep inference-time VGG-like (e.g., branch-free) model and constructed extra ResNet-style shortcuts parallel. After training, these residual kernels are absorbed into a single kernel of size 3. In this paper, we use this methodology to fuse short kernels after training. Consequently, we make the long convolution kernel capable of capturing flexible patterns, improving overall performance.

4.3. CHELA Model

The gating mechanism in CHELA is based on GRU (Cho et al., 2014) and GAU (Hua et al., 2022) as the backbone architectures, with the output of short-long convolutions layer $\mathbf{Z}$ embedded into the linear attention matrix. $\mathbf{Z}$ can be considered a contextual abstraction because it provides global information through multi-level structured patterns:

$$\mathbf{M}_{\text{linear}} = \text{Norm}(\mathbf{Q}(\mathbf{K}^\top\mathbf{V})) \odot \mathbf{G}_a \tag{9}$$
$$\mathbf{G}_a = \phi_{\text{silu}}(\mathbf{Z}\mathbf{W}_v + b_g) \tag{10}$$
$$\mathbf{Q} = \alpha_q \odot \mathbf{Z} + \beta_q \tag{11}$$
$$\mathbf{K} = \alpha_k \odot \mathbf{Z} + \beta_k \tag{12}$$
$$\mathbf{V} = \phi_{\text{silu}}(\mathbf{X}\mathbf{W}_v + b_v) \tag{13}$$

where $\alpha_q, \alpha_k, \beta_q, \beta_k \in \mathbb{R}^d$ are the learnable scalers and offsets of queries and keys like GAU, respectively. The $\mathbf{M}_{\text{linear}}$ is the output of flash linear attention with gate $\mathbf{G}_a$. Moreover, we employ an output gate $\mathbf{G}_o$ to update final activation output $\mathbf{U}$ in a residual style:

$$\mathbf{G}_o = \phi_{\text{sigmoid}}(\mathbf{Z}\mathbf{W}_o + b_o) \tag{14}$$
$$\mathbf{U} = \mathbf{M}_{\text{linear}} \odot \mathbf{G}_o + \mathbf{X} \odot (1 - \mathbf{G}_o) \tag{15}$$

the architecture of CHELA is demonstrated in Fig. 2 *(a)*.

CHELA Block Besides, each standard CHELA block is also equipped with a two-layer feedforward neural network, layer normalization (Ba et al., 2016), and residual connections (He et al., 2016) as channel mixing. Concretely, given the input $\mathbf{X}$ to the CHELA block, we have the output $\mathbf{Y}$ as:

$$\mathbf{X}_a = \text{CHELA}\left(\text{LayerNorm}(\mathbf{X})\right)$$
$$\mathbf{Y} = \text{FFN}\left(\text{LayerNorm}(\mathbf{X}_a)\right) + \mathbf{X}_a.$$

Note that here we apply pre-layer normalization, similar to Vaswani et al. (2017). There are other works (e.g., Devlin et al. 2019) that apply post-layer normalization, where the normalization is applied after the residual connection.

4.4. Relation to MEGA and SPADE.

As hybrid models of SSM and attentions, we adhered to the idea of proposing an efficient variant to utilize both well-structured and fully data-dependent patterns on long-sequence modeling. However, it is undeniable that the linear versions of Mage-chunk and SPADE-chunk have caused significant performance degradation, which means they struggle to make good tradeoffs between complexity and performance. The EMA Mega uses is too simple to capture higher-order information, so the overall performance is heavily dependent on attention. SPADE uses a parallel structure that directly combines the SSM and Attention modules, which reduces the dependence on attention, but the linear strategy of dividing the chunks still limits the model's performance. CHELA maximizes efficiency by enhancing structured representations with short-length convolutions and lossless use of linear attention in a hardware-optimized implementation. *Ultimately,* CHELA *boosting performance while also dramatically increasing speed in linear time.*

5. Experiments

To assess CHELA, we performed tests on five standard sequence modeling tasks involving different types of data, and we compared them with the latest cutting-edge models for each task. All experiments were realized based on NVIDIA A100-80G and Pytorch. We used float32 parameters, with bfloat16 precision for most computations.

5.1. Long-Context Sequence Modeling

We conducted experiments to evaluate sequence models using the Long Range Arena (LRA) benchmark. This benchmark, introduced by (Tay et al., 2020b), is designed to assess the performance of sequence models in long-context learning. The benchmark includes six tasks, namely ListOps (Nangia & Bowman, 2018), byte-level Text classification (Maas et al., 2011), byte-level document Retrieval (Radev et al., 2013), Image classification on sequences of pixels (Krizhevsky et al., 2009), Pathfinder (Lins-

Table 1. Performance of predicting outcomes of list operations in the LRA (Tay et al., 2020b). Bold indicates the best-performing model and underlines the second best. Results are taken from either the citation. The training speed and peak memory consumption comparison on the Text task with an input length of 4K.

Models	ListOps	Text	Retrieval	Image	Pathfinder	PathX	Avg.	Speed
Attention:								
Transformer (Vaswani et al., 2017)	36.37	64.27	57.46	42.44	71.40	✗	54.39	1.0×
Local Attention (Tay et al., 2020b)	15.82	63.98	52.98	41.46	66.63	✗	46.06	5.3×
Linear Trans. (Katharopoulos et al., 2020)	16.13	65.90	53.09	42.34	75.30	✗	50.55	4.7×
Linformer (Wang et al., 2020)	35.70	53.94	52.27	38.56	76.34	✗	51.36	5.5×
Sparse Transformer (Child et al.)	17.07	63.58	59.59	44.24	71.71	✗	51.24	4.2×
Performer (Choromanski et al., 2020)	18.01	65.40	53.82	42.77	77.05	✗	51.41	5.7×
Sinkhorn Transformer (Tay et al., 2020a)	33.67	61.20	53.83	41.23	67.45	✗	51.39	3.8×
Longformer (Beltagy et al., 2020)	35.63	64.02	59.29	40.83	74.87	✗	55.01	1.1×
BigBird (Zaheer et al., 2020)	36.05	64.02	59.29	40.83	74.87	✗	55.01	1.1×
Luna-256 (Ma et al., 2021)	37.25	65.78	79.56	47.86	78.55	✗	61.95	4.9×
Reformer (Kitaev et al., 2020)	37.27	56.10	53.40	38.07	68.50	✗	50.67	0.8×
State Space Models:								
S4 (Gu et al., 2022)	59.60	86.82	76.02	87.09	87.26	86.26	80.50	-
DSS (Gupta et al., 2022)	57.60	59.60	86.82	90.90	88.65	94.20	86.09	4.8×
S4D (Gu et al., 2022)	60.18	87.34	91.09	87.83	93.78	92.80	85.50	-
Mamba (Gu & Dao, 2023)	38.02	82.98	72.14	87.83	69.82	67.32	66.59	-
Linear Hybrid:								
Mega-chunk (Ma et al., 2022)	58.76	90.19	90.97	85.80	94.41	93.81	85.66	5.5×
SPADE-chunk (Zuo et al., 2023)	60.50	90.69	91.17	88.22	96.23	97.60	87.40	5.5×
CHELA (Ours)	**60.69**	**91.10**	**91.65**	**91.12**	**96.40**	**98.15**	**88.19**	**5.8×**

ley et al., 2018), and its extreme long version (Path-X; Tay et al. (2020b)). The input sequences for these tasks range from 1K to 16K tokens and cover various data modalities. In Table 1, CHELA is compared to various baselines, such as Transformer and its efficient versions, as well as the top-performing S4 models. In order to make a fair comparison, we make sure that Mega and S4 have a similar number of parameters by balancing the number of layers and model sizes for each task. The results are based on the average of five runs with different random seeds, and you can find the tuning information and model details in the Appendix. The performance of our model has been outstanding across all six tasks, achieving an average accuracy of 88.26% and surpassing all the other comparison methods. Additionally, we assessed the speed of our model when applied to the byte-level classification task with a 4K input. Our hardware-efficient linear mechanism has demonstrated remarkable efficiency, with a speed that is 5.8 times faster. It is important to highlight that our model, with its unique short-long convolutions hybrid design, exhibits even greater efficiency compared to a variety of linear Transformers, Structured State Space Models, and recent hybrid models.

Table 2. Accuracy on Speech Commands dataset.

Model	SpeechCommand-Raw #Param.	Accuracy
Transformer	786K	✗
S4 (Gu et al., 2021)	300K	97.50
Mega (Ma et al., 2022)	-	✗
Mega-chunk (Ma et al., 2022)	476K	96.03
CHELA (ours)	493K	**97.98**

5.2. Raw Speech Classification

We intend to evaluate the capability of CHELA in modeling lengthy speech signals by employing it for the classification of unaltered speech signals with a duration of 16000, instead of depending on traditional preprocessing techniques like converting them into MFCC features. As per Gu et al. (2021) approach, we classify speech on the Speech Commands dataset's SC10 subset, which was introduced by Warden (2018). As reported in (Ma et al., 2022), the Mega-chunk uses a chunk size of 1000 to enable processing the data. In Table 2, our model has 493K parameters and achieves a 97.98% accuracy, making it the leading method in this table. This result is primarily due to the suitability of long convolutions for processing the numerous continuous and low-frequency signals present in speech. Additionally, the ability of short convolutions to capture rich global information enables attention to focus on important aspects.

5.3. Auto-Regressive Language Modeling

By following Ma et al. (2022); Lingle (2023), we assess CHELA on two popular language modeling datasets, *i.e.*, WikiText-103 (Merity et al., 2016) and enwik8 (Mahoney, 2011), which are next-token prediction tasks. WikiText-103 is a dataset for word-level language modeling with 103 million tokens from Wikipedia articles in its training set. In line with previous work (Baevski & Auli, 2019), our method involves using adaptive softmax and input embeddings, and we utilize a vocabulary of 260,000 tokens. Enwik8 is a commonly used benchmark for character-level language

Table 3. Performance of pixel-level classification on the sCIFAR.

Model	Accuracy (%)
Attention:	
Transformer (Trinh et al., 2018)	62.20
RNN:	
LSTM (Hochreiter & Schmidhuber, 1997)	63.01
r-LSTM (Trinh et al., 2018)	72.20
UR-GRU (Gu et al., 2020b)	74.40
HiPPO-RNN (Gu et al., 2020a)	61.10
LipschitzRNN (Erichson et al., 2020)	64.20
State Space Models:	
S4 (Gu et al., 2022)	91.80
S4D (Gu et al., 2022)	90.69
S5 (Smith et al., 2023)	90.10
Liquid-S4 (Hasani et al., 2022)	92.02
Convolution:	
TrellisNet (Bai et al., 2018)	73.42
CKConv (Li et al., 2022)	63.74
FlexConv (Romero et al., 2021)	80.82
MultiresNet (Shi et al., 2023)	93.15
CHELA (ours)	**94.02**

modeling, presenting a significant challenge to models. It comprises approximately 100 million unprocessed tokens from Wikipedia articles and has a vocabulary size of about 200. When evaluating language models, we segment the test data and process each segment sequentially during testing to assess their effectiveness. In Table 4, we compare with previous top-performing models that are designed to take advantage of longer context, including Transformers (Baevski & Auli, 2019), Transformer-XL and S4 (Gu et al., 2021). The model we developed demonstrated outstanding performance on both WikiText-103 and enwik8 datasets, outperforming the baseline models by a significant margin. Our model achieves an inference speed that is almost 10 times faster than the Pure Transformer model. The hybrid structure of the short-long convolutions layer plays a crucial role in enabling our model to manage length extrapolation during inference, allowing it to process longer sequences than those encountered during training. This distinctive characteristic of our model enhances its capability to naturally handle complex tasks, making it a valuable addition to any long-sequence project.

Table 4. Performance and training speed on WikiText-103 dataset.

	WikiText-103		
Model	**#Param.**	**PPL**	**Speed**
Transformer-adaptive	247M	18.66	5.6 k t/s
Transformer-XL	257M	18.30	-
S4(Gu et al., 2020b)	249M	20.95	-
Mega-chunk(Ma et al., 2022)	252M	18.07	48k t/s
CHELA (ours)	258M	**16.97**	53k t/s

Table 5. Testing bits-per-byte on Enwik8 dataset.

	enwik8	
Model	**#Param.**	**PPL**
Transformer-XL	41M	1.06
Mega (Ma et al., 2022)	39M	1.02
Transformer-VQ (Lingle, 2023)	190M	0.99
CHELA (ours)	48M	**0.96**

5.4. Pixel-Level Sequential Image Classification

Begin by addressing tasks related to image classification, in which images are considered as a one-dimensional sequence of pixels. In these tasks, models cannot rely on preconceived two-dimensional structures within the image. Consequently, the model must possess the ability to recognize patterns at different temporal scales, including pixels that are close together in the original image but far apart in their sequential representation. We evaluate the performance of our model using the Sequential CIFAR-10 dataset, commonly used as a benchmark for capturing long-term dependencies in RNNs. The CIFAR-10 dataset is frequently employed in machine learning for tasks on image classification. Within this dataset, the typical training and testing split is maintained, reserving 10% of the training set for validation purposes. To categorize the images, the mean of all tokens in the output sequences is computed, and the resulting values are subjected to a fully connected layer to produce class logits. The Table 3 displays the results. CHELA has achieved state-of-the-art performance and the best test accuracy on the sequence classification task, surpassing multiple strong competitors such as Transformers (Vaswani et al., 2017), RNNs, state space models, and other convolutional models. In particular, the CHELA model has exceeded the performance of previous convolution-based models by more than ten percentage points. It is important to note that our model has delivered impressive results by surpassing the previously established performance standard, even though it uses a relatively simple architecture. The model primarily employs a hybrid method that compresses long historical information based on the output of short-long convolutions. Our most effective model consists of ten CHELA blocks, which significantly contribute to achieving exceptional performance.

6. Ablation Study

Our ablation experiments focus on answering two key questions mostly related to our design: (1) Does the hardware-friendly implementation significantly improve the speed of linear attention? (2) The effectiveness of our proposed short-long convolutions module on long sequences.

Q1: Benchmark hardware-efficient linear attention. To answer the first question, our Hardware-Efficient Linear

Attention achieves almost real linear relationships with sequence lengths. We conducted an analysis on the WikiText-103 dataset with models with 200M parameters. As visualized in Fig. 4, we have more than doubled the speedup of the original Pytorch implementation of the linear attention.

Q2: Analysis of short-long convolutions. To answer the second question, we further combined a variety of hybrid models following the modeling structure of CHELA. Specifically, we compared the representative SSM-like modules on the subset of the LRA (Tay et al., 2020b) dataset (Text, Image, and PathX). It is clear that the proposed Short-Long Convolutions are the best partner for linear attention.

Table 6. Ablation study on different structured mixers in CHELA.

Methods	Datasets		
	Text	Image	PathX
Damped EMA (Ma et al., 2022)	90.19	85.80	93.81
S4D (Gu et al., 2022)	90.85	88.95	94.29
Long Conv (Fu et al., 2023b)	90.35	87.57	97.24
Short-Long Convs	91.10	91.12	98.65

7. Related Works

Efficient transformer models A variety of efforts have been made to decrease the quadratic time and space complexity of standard attention mechanisms. One method is to utilize "sparse attention," where each token only attends to a subset of all the tokens based on predefined patterns, such as neighboring tokens within a fixed-size window. (Child et al., 2019) started the attempt to sparse the attention, and then there were a lot more followers, such as ETC (Ainslie et al., 2020), Longformer (Beltagy et al., 2020), BigBird (Zaheer et al., 2020), Poolingformer (Zhang et al., 2021), and HEPOS (Huang et al., 2021) are some examples of this approach. Another option is to utilize "low-rank projection," as mentioned in the work by (Wang et al., 2020). Similar techniques include Nyströmformer (Xiong et al., 2021), Synthesizer (Tay et al., 2021), and Luna (Ma et al., 2021). However, these methods encounter challenges when dealing with causal tasks, such as auto-regressive language modeling. Another approach uses "clustering method," where we partition $\mathbf{Q}$ or $\mathbf{K}$ into multiple clusters and perform inter-cluster attention. Examples of such methods include Sinkhorn Transformer (Tay et al., 2020a), Reformer (Kitaev et al., 2020), Routing Transformer (Roy et al., 2021), and simplified FLASH (Hua et al., 2022), etc. "Methods based on kernels" can be utilized to approximate the complete attention $\mathrm{Attn}(\mathbf{X})$. These methods replace the quadratic-time softmax attention with fast linear-time kernel approximations (such as Gaussian and arc-cosine kernels). Some instances of this approach include Linear Transformer (Katharopoulos et al., 2020), Performer (Choromanski et al., 2020), and FMMformer (Nguyen et al., 2021), etc. Both low-dimensional projection and methods based on kernels are employed to estimate full attention and, as a result, are vulnerable to significant approximation errors.

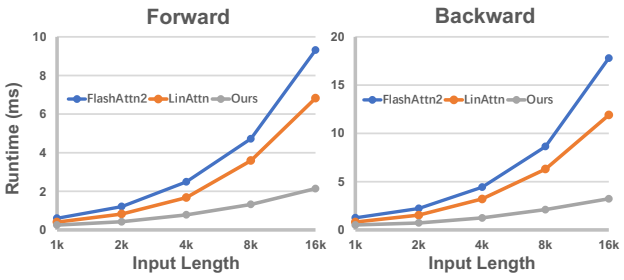

Figure 4. Comparative Analysis of Speed: Runtime in milliseconds for the forward and backward pass across varying lengths.

State space models and long convolutions Recurrent neural networks and their linear counterparts such as state-space models are capable of retaining memory of the past. Among these models, S4 (Gu et al., 2021) is notable because it can be implemented through convolutions thanks to its linear recurrence. However, the long convolution kernel for this model is as long as the input sequence, and its efficient computation requires sophisticated parameterization and approximation techniques. Although recent advances have found solutions to this issue, initializing these models still requires special effort (Gupta et al., 2022; Gu et al., 2020b). Many of these models use the HiPPO (Gu et al., 2020a) initialization mechanism, which aims to memorize historical data through projection to orthogonal polynomials. Based on a structure similar to SSM, an increasing number of models focusing on either linear recurrence or global convolution have been developed recently (Fu et al., 2023b;a; Poli et al., 2023; Gu & Dao, 2023).

Hardware-efficient implementation The FlashAttention series (Dao et al., 2022; Dao, 2023) is dedicated to optimizing the standard attention operator for GPU platforms at the system level. Its effectiveness has been extensively validated. The approach involves using tiling strategies to minimize the amount of memory reads/writes between the high bandwidth memory (HBM) and on-chip SRAM.

8. Conclusion and Limitations

We presented CHELA for robust and efficient modeling of long sequences. CHELA is an SSM-attention hybrid architecture that computes both structured and data-dependent patterns in linear time with respect to sequence length. Its superior performance is enabled by considering the global view of the Short-Long Convolutions and real linear

Hardware-Efficient Linear Attention with gating mechanisms. Compared to other hybrid linear models, we are currently the first to achieve performance improvement while maintaining linear complexity. Our large-scale and diverse experiments demonstrate that CHELA is an efficient and flexible long sequence model with excellent performance on image, text, logical reasoning, and speech data.

However, CHELA also has some limitations. The optimal combinations of short convolutions are not explored in this paper, which should be a future research direction of designing dynamic short-convolutional components according to the input data. Moreover, the time-varying SSM is a different idea to achieve this goal. By embracing CHELA as a starting point in the integration of hardware-efficient implementation into hybrid models, we are taking the initial step towards achieving greater efficiency gains in the future.

Acknowledgements

This work was supported by the Science & Technology Innovation 2030 Major Program Project No. 2021ZD0150100, National Natural Science Foundation of China Project No. U21A20427, Project No. WU2022A009 from the Center of Synthetic Biology and Integrated Bioengineering of Westlake University, and Project No. WU2023C019 from the Westlake University Industries of the Future Research. Finally, we thank the Westlake University HPC Center for providing part of the computational resources.

Impact Statement

The goal of this paper is to advance research in long-sequence modeling by introducing an efficient model design CHELA. We have considered broader ethical impacts and do not foresee CHELA directly leading to negative societal consequences. All datasets and models used are existing public resources that do not contain private or sensitive information. Through discussing the hybrid design of linear attention and SSM models, we aim to make sequence mixers much more efficient by fully leveraging hardware and different model features. Besides, as the community proposes new methods, we encourage discussing any potential negative impacts early in the research process. Overall, we believe hardware-efficient style and the combination of structured and data-dependent patterns are the dominant trends of the future for efficient sequence modeling.

References

Ainslie, J., Ontanon, S., Alberti, C., Cvicek, V., Fisher, Z., Pham, P., Ravula, A., Sanghai, S., Wang, Q., and Yang, L. ETC: Encoding long and structured inputs in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 268–284, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.19. URL https://aclanthology.org/2020.emnlp-main.19.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *ArXiv preprint*, abs/1607.06450, 2016. URL https://arxiv.org/abs/1607.06450.

Baevski, A. and Auli, M. Adaptive input representations for neural language modeling. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=ByxZX20qFQ.

Bai, S., Kolter, J. Z., and Koltun, V. Trellis networks for sequence modeling. *arXiv preprint arXiv:1810.06682*, 2018.

Beltagy, I., Peters, M. E., and Cohan, A. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Child, R., Gray, S., Radford, A., and Sutskever, I. Generating long sequences with sparse transformers. arxiv 2020. *arXiv preprint arXiv:1904.10509*.

Child, R., Gray, S., Radford, A., and Sutskever, I. Generating long sequences with sparse transformers. *ArXiv preprint*, abs/1904.10509, 2019. URL https://arxiv.org/abs/1904.10509.

Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches, 2014.

Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, L., et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

Dao, T. Flashattention-2: Faster attention with better parallelism and work partitioning, 2023.

Dao, T., Fu, D. Y., Ermon, S., Rudra, A., and Ré, C. Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Ding, X., Zhang, X., Zhou, Y., Han, J., Ding, G., and Sun, J. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns, 2022.

Ding, X., Zhang, Y., Ge, Y., Zhao, S., Song, L., Yue, X., and Shan, Y. Unireplknet: A universal perception large-kernel convnet for audio, video, point cloud, time-series and image recognition, 2023.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

Erichson, N. B., Azencot, O., Queiruga, A., Hodgkinson, L., and Mahoney, M. W. Lipschitz recurrent neural networks. *arXiv preprint arXiv:2006.12070*, 2020.

Fu, D. Y., Dao, T., Saab, K. K., Thomas, A. W., Rudra, A., and Ré, C. Hungry hungry hippos: Towards language modeling with state space models, 2023a.

Fu, D. Y., Epstein, E. L., Nguyen, E., Thomas, A. W., Zhang, M., Dao, T., Rudra, A., and Ré, C. Simple hardware-efficient long convolutions for sequence modeling, 2023b.

Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces, 2023.

Gu, A., Dao, T., Ermon, S., Rudra, A., and Ré, C. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33: 1474–1487, 2020a.

Gu, A., Gulcehre, C., Paine, T., Hoffman, M., and Pascanu, R. Improving the gating mechanism of recurrent neural networks. In *International Conference on Machine Learning*, pp. 3800–3809. PMLR, 2020b.

Gu, A., Goel, K., and Ré, C. Efficiently modeling long sequences with structured state spaces. *ArXiv preprint*, abs/2111.00396, 2021. URL https://arxiv.org/abs/2111.00396.

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

Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.

Gupta, A., Gu, A., and Berant, J. Diagonal state spaces are as effective as structured state spaces. *Advances in Neural Information Processing Systems*, 35:22982–22994, 2022.

Hasani, R., Lechner, M., Wang, T.-H., Chahine, M., Amini, A., and Rus, D. Liquid structural state-space models. *arXiv preprint arXiv:2209.12951*, 2022.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

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

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.

Huang, L., Cao, S., Parulian, N., Ji, H., and Wang, L. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1419–1436, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.112. URL https://aclanthology.org/2021.naacl-main.112.

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

Kitaev, N., Kaiser, Ł., and Levskaya, A. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Li, S., Wang, Z., Liu, Z., Tan, C., Lin, H., Wu, D., Chen, Z., Zheng, J., and Li, S. Z. Efficient multi-order gated aggregation network, 2023.

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

Lin, T., Wang, Y., Liu, X., and Qiu, X. A survey of transformers. *AI Open*, 2022.

Lingle, L. D. Transformer-vq: Linear-time transformers via vector quantization, 2023.

Linsley, D., Kim, J., Veerabadran, V., Windolf, C., and Serre, T. Learning long-range spatial dependencies with horizontal gated recurrent units. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 152–164, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/ec8956637a99787bd197eacd77acce5e-Abstract.html.

Liu, Z., Li, S., Wu, D., Liu, Z., Chen, Z., Wu, L., and Li, S. Z. Automix: Unveiling the power of mixup for stronger classifiers, 2022.

Liu, Z., Li, S., Wang, G., Tan, C., Wu, L., and Li, S. Z. Harnessing hard mixed samples with decoupled regularizer, 2023.

Ma, X., Kong, X., Wang, S., Zhou, C., May, J., Ma, H., and Zettlemoyer, L. Luna: Linear unified nested attention. *Advances in Neural Information Processing Systems*, 34: 2441–2453, 2021.

Ma, X., Zhou, C., Kong, X., He, J., Gui, L., Neubig, G., May, J., and Zettlemoyer, L. Mega: Moving average equipped gated attention. *ArXiv preprint*, abs/2209.10655, 2022. URL https://arxiv.org/abs/2209.10655.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1015.

Mahoney, M. Large text compression benchmark, 2011. URL http://www.mattmahoney.net/dc/text.html.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models, 2016.

Nangia, N. and Bowman, S. ListOps: A diagnostic dataset for latent tree learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pp. 92–99, New Orleans, Louisiana, USA, 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-4013. URL https://aclanthology.org/N18-4013.

Nguyen, T., Suliafu, V., Osher, S., Chen, L., and Wang, B. Fmmformer: Efficient and flexible transformer via decomposed near-field and far-field attention. *Advances in Neural Information Processing Systems*, 34, 2021.

Poli, M., Massaroli, S., Nguyen, E., Fu, D. Y., Dao, T., Baccus, S., Bengio, Y., Ermon, S., and Ré, C. Hyena hierarchy: Towards larger convolutional language models, 2023.

Qin, Z., Li, D., Sun, W., Sun, W., Shen, X., Han, X., Wei, Y., Lv, B., Luo, X., Qiao, Y., and Zhong, Y. Transnormerllm: A faster and better large language model with improved transnormer, 2024.

Radev, D. R., Muthukrishnan, P., Qazvinian, V., and Abu-Jbara, A. The acl anthology network corpus. *Language Resources and Evaluation*, 47(4):919–944, 2013.

Ramachandran, P., Zoph, B., and Le, Q. V. Searching for activation functions, 2017.

Romero, D. W., Bruintjes, R.-J., Tomczak, J. M., Bekkers, E. J., Hoogendoorn, M., and van Gemert, J. C. Flexconv: Continuous kernel convolutions with differentiable kernel sizes. *arXiv preprint arXiv:2110.08059*, 2021.

Roy, A., Saffar, M., Vaswani, A., and Grangier, D. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021. doi: 10.1162/tacl_a_00353. URL https://aclanthology.org/2021.tacl-1.4.

Shi, J., Wang, K. A., and Fox, E. B. Sequence modeling with multiresolution convolutional memory, 2023.

Smith, J. T., Warrington, A., and Linderman, S. W. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*, 2023.

Tay, Y., Bahri, D., Yang, L., Metzler, D., and Juan, D.-C. Sparse sinkhorn attention. In *International Conference on Machine Learning*, pp. 9438–9447. PMLR, 2020a.

Tay, Y., Dehghani, M., Abnar, S., Shen, Y., Bahri, D., Pham, P., Rao, J., Yang, L., Ruder, S., and Metzler, D. Long range arena: A benchmark for efficient transformers. *arXiv preprint arXiv:2011.04006*, 2020b.

Tay, Y., Bahri, D., Metzler, D., Juan, D., Zhao, Z., and Zheng, C. Synthesizer: Rethinking self-attention for transformer models. In Meila, M. and Zhang, T. (eds.), *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*, pp. 10183–10192. PMLR, 2021. URL http://proceedings.mlr.press/v139/tay21a.html.

Trinh, T., Dai, A., Luong, T., and Le, Q. Learning longer-term dependencies in rnns with auxiliary losses. In *International Conference on Machine Learning*, pp. 4965–4974. PMLR, 2018.

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.

Wang, S., Li, B. Z., Khabsa, M., Fang, H., and Ma, H. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Warden, P. Speech commands: A dataset for limited-vocabulary speech recognition, 2018.

Xiong, Y., Zeng, Z., Chakraborty, R., Tan, M., Fung, G., Li, Y., and Singh, V. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 14138–14148, 2021.

Yang, S., Wang, B., Shen, Y., Panda, R., and Kim, Y. Gated linear attention transformers with hardware-efficient training, 2024.

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

Zhang, H., Gong, Y., Shen, Y., Li, W., Lv, J., Duan, N., and Chen, W. Poolingformer: Long document modeling with pooling attention. In Meila, M. and Zhang, T. (eds.), *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*, pp. 12437–12446. PMLR, 2021. URL http://proceedings.mlr.press/v139/zhang21h.html.

Zuo, S., Liu, X., Jiao, J., Charles, D., Manavoglu, E., Zhao, T., and Gao, J. Efficient long sequence modeling via state space augmented transformer, 2023.

A. Experimental Details

A.1. Long Range Arena (LRA) and sCIFAR

For all tasks, we closely follow Tay et al. (2020b) for details such as data preprocessing, data split, etc. The hyper-parameters of CHELA models on these tasks are listed in Table 7. The experimental configuration of sCIFAR follows the parameter settings of the image in LRA.

Table 7. Hyper-parameters of CHELA models on LRA and raw speech classification tasks. BSZ is batch size, LR is learning rate and WD is weight decay. BN, LN and SN refer to Batch Normalization, Layer Normalization and Scale Normalization.

Task	Depth	d_{model}	d_{FFN}	Attn-FN	Norm	Pre-norm	BSZ	LR	Dropout	WD	Epochs
ListOps	6	80	160	laplace	BN	False	64	0.001	0.1	0.01	60
Text	4	128	256	norm	SN	False	50	0.004	0.1	0.01	50
Retrieval	6	128	256	norm	SN	False	64	0.003	0.1	0.04	40
Image	8	160	320	laplace	BN	True	50	0.01	0.0	0.02	200
Pathfinder	6	128	256	laplace	BN	True	128	0.01	0.0	0.01	200
Path-X	4	64	128	laplace	BN	True	128	0.01	0.0	0.01	100
SC	6	60	120	laplace	BN	True	20	0.01	0.0	0.01	200

A.2. Language Modeling

We use the data of WikiText-103 and enwik8 and their splits provided by Ma et al. (2022). At training time, we split the training data into segments; each segment contains m consecutive chunks, where the chunk size is the effective attention length. Other training hyperparameters, including optimizer, learning rate scheduler, and architecture, are presented in Table 8.

Table 8. Hyper-parameters of models for language modeling.

	WikiText-103	**enwik8**
Batch Size $\times$ GPUs	6144 $\times$ 24	8192 $\times$ 8
Optimizer	AdamW	AdamW
Learning Rate	0.005	0.005
Adam-β	$(0.9, 0.98)$	$(0.9, 0.98)$
Learning Rate Decay	linear	linear
Weight Decay	0.1	0.1
Dropout	0.3	0.1
Attention Dropout	0.1	0.0
FFN Hidden Dropout	0.1	0.0
Gradient Clipping	1.0	1.0
Warmup steps	24K	24K
Total updates	400K	400K
Decoder Layers	16	12
Model size	1024	512
FFN Hidden size	1536	1024
Total Parameters	258M	48M