Focus Your Attention (with Adaptive IIR Filters)

Shahar Lutati and **Itamar Zimerman** and **Lior Wolf**
The School of Computer Science, Tel Aviv University
shahar761@gmail.com
zimerman1@mail.tau.ac.il
wolf@cs.tau.ac.il

Abstract

We present a new layer in which dynamic (i.e., input-dependent) Infinite Impulse Response (IIR) filters of order two are used to process the input sequence prior to applying conventional attention. The input is split into chunks, and the coefficients of these filters are determined based on previous chunks to maintain causality. Despite their relatively low order, the causal adaptive filters are shown to focus attention on the relevant sequence elements. The new layer is grounded in control theory, and is shown to generalize diagonal state-space layers. The layer performs on-par with state-of-the-art networks, with a fraction of their parameters and with time complexity that is sub-quadratic with input size. The obtained layer is favorable to layers such as Heyna, GPT2, and Mega, both with respect to the number of parameters and the obtained level of performance on multiple long-range sequence problems.

1 Introduction

Designing sequence models that capture short- and long-term dependencies is a central goal of sequence modeling. Besides performance, computational complexity also plays a part when dealing with long sequences. Although transformers (Vaswani et al., 2017) excel at tasks that involve short-range dependencies, their performance on data with long-range dependencies can be poor. For example, regardless of the high time complexity, on the Long Range Arena benchmark (LRA) (Tay et al., 2020) transformers perform poorly compared to other sequence models.

Another approach that has emerged to address long-range data processing is the utilization of regularized implicit global (long) convolutions. In this technique, convolutions are employed along the sequence dimension, enabling convolutions with a global receptive field. Initially, this approach was implemented through state-space layers (Gu et al., 2021b,a), which introduced a recurrent layer that could be efficiently computed via a global convolution. Recent research has explored alternative variations of global convolutions, including implicit (Poli et al., 2023) and regularized parameterization (Fu et al., 2023a; Li et al., 2022). These methods have demonstrated improved performance on tasks involving long-range dependencies and with sub-quantitative complexity. They have also shown effectiveness in enhancing long-range transformer capabilities (Ma et al., 2022; Zuo et al., 2022; Saon et al., 2023). However, it remains uncertain whether these models can scale up or function similarly to transformers across a diverse range of tasks (Vardasbi et al., 2023).

This work strives to efficiently integrate convolution-based sequence models and transformers, to provide a model that is capable of handling both short and long dependencies. The attempt to combine these components was first presented by Ma et al. (2022), who used a simple global convolution before each transformer block. This convolution is parameterized by the Exponential Moving Average (EMA) recurrent rule and can be seen as an IIR filter. In this work, instead of using first-order IIR filters, we introduce learnable adaptive IIR filters, which allow us to propose Focus, a layer that combines local attention and a novel type of regularized global convolution grounded on a hypernetwork that produces adaptive IIR filters.

Our main contribution is the focus layer, which has several unique properties: (i) We are the first to use data-dependent global filters, which are implemented by a global hyper-network mechanism that focuses local attention. (ii) In contrast to other works in the domain that employ FIR filters, it relies on IIR filters. (iii). We present an efficient and stable computation of those IIR filters. (iv) Theoretically, our layer is grounded in the theory of control systems, similar to state-space layers, which are built on the state-space model (SSM) of control theory. Furthermore, in Sec. 4 we show

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 12538–12549
December 6-10, 2023 ©2023 Association for Computational Linguistics

that IIR filters are a generalization of SSMs and diagonal-linear RNNs, which have recently been recognized as remarkable long-range learning architectures (Gupta et al., 2022a; Gu et al., 2022; Orvieto et al., 2023; Gupta et al., 2022b; Saon et al., 2023; David et al., 2023). By drawing upon the extensive research conducted on IIR filters, our findings can provide additional insights into the effectiveness, stability, expressiveness, and initialization of those layers.

2 Background and related work

IIR filters, known as infinite impulse response filters, are digital filters that utilize feedback to generate an output signal. Their primary applications involve signal smoothing, filtering, and signal modification. These filters are extensively employed in various fields, such as audio processing, speech processing, and image processing. One notable advantage of IIR filters is their ability to achieve a significantly sharper roll-off in the transition region compared to an FIR filter of the same order. This is made possible by the presence of complex poles in the IIR filters, which enable them to attenuate frequencies more rapidly.

The state-space representation of an IIR filter is a convenient way to represent the filter's dynamics and to implement it in software. It consists of three parts: the state vector, the state transition matrix, and the output matrix. The state vector contains the filter's internal state variables, the state transition matrix describes how the state vector changes over time, and the output matrix describes how the output signal is computed from the state vector. Such representation are described in (Zhang et al., 2023)

Learnable IIR Filters Since IIR filters are computationally efficient, yet expressive, it is natural to design IIR filters with Deep Learning. (Kuznetsov et al., 2020) proposes an approach to using traditional digital IIR filter structures inside deep-learning networks trained using backpropagation. The authors establish the link between such structures and recurrent neural networks and present three different differentiable IIR filter topologies. They compare the proposed topologies against each other and an established baseline and show that the proposed topologies can achieve better performance in some cases. Additionally, the authors present a simple Wiener-Hammerstein model, using differentiable IIRs as its filtering component, and train it on a guitar signal.

Global Convolutions The global convolution, also known as a long convolution, is a layer that applies scalar convolutions along the sequence dimension, enabling the handling of unrestricted 1-D sequences. Empirically, these layers have shown strong performance in tasks involving long-range dependencies, particularly in domains such as NLP (Dao et al., 2022b; Mehta et al., 2022; Wang et al., 2022), audio (Goel et al., 2022), speech (Saon et al., 2023), video (Islam et al., 2022; Wang et al., 2023), time-series analysis (Zhang et al., 2023) and more. Moreover, they exhibit computational efficiency as their cost is sub-quadratic. However, to achieve SOTA results, appropriate regularization is necessary. The approach of (Gu et al., 2021b,a; Ma et al., 2022; Li et al., 2022) incorporates a parameterization that inherently regularizes the kernel and decoupling sequence length from parameter count. (Romero et al., 2021; Poli et al., 2023) utilizes an implicit parameterization learned by FFNs operating on positional encodings, while (Fu et al., 2023a) explicitly regularizes the convolution kernels using squash or smooth operators.

Long Range Transformers Transformers (Vaswani et al., 2017) have emerged as highly effective models for various tasks, but their widespread adoption has been limited by the quadratic cost of the self-attention mechanism and poor performance on long-range tasks. Researchers have pursued diverse approaches to overcome this challenge and to create efficient transformer architectures (Fournier et al., 2021; Tay et al., 2022). From the perspective of efficiency, techniques such as sparse attention (Child et al., 2019), low-rank attention (Wang et al., 2020; Winata et al., 2020), kernel-based attention (Choromanski et al., 2020), recurrent mechanisms (Hutchins et al., 2022; Dai et al., 2019), and efficient IO-awareness-based implementation (Dao et al., 2022a) proved efficient. From the perspective of effectiveness, (Yu et al., 2023; Ivgi et al., 2023) combines local and global attention models hierarchically, enhancing the model's ability to handle extensive context Other techniques employ global memory-based Attention (Gupta and Berant, 2020; Al Adel, 2022; Al Adel and Burtsev, 2021), and (Zhou et al., 2022) applies attention in the frequency domain to expand long-range capabilities.

Hyper Networks A hypernetwork (Ha et al., 2016) is a function that maps a set of inputs to a set of weights, which are used as the parameters of a "primary network". Hypernetworks have been shown to be effective for a variety of tasks, including, for example, image classification (Lutati and Wolf, 2023), natural language processing (He et al., 2022), and speech recognition (Szatkowski et al., 2022). They have also been shown to be able to improve the performance of neural networks on meta-learning tasks, such as few-shot learning (Bertinetto et al., 2016), continual learning (Von Oswald et al., 2019), and neural architecture search (Zhang et al., 2019).

Adaptive Filtering Adaptive filtering is a technique used to improve the quality of a signal by removing noise or interference. Adaptive filters are able to adapt to changes in the signal or the environment, making them well-suited for a variety of applications.

One common technique used in adaptive filtering is the short-time Fourier transform (STFT), which provides a time-frequency representation of a signal. It enables the analysis of time-varying properties of a signal by dividing it into short-time windows and applying the Fourier transform to each window. The STFT reveals the distribution of frequency content over time, which allows the adaptive filter to track the frequency content of the signal and adapt its coefficients accordingly. However, STFT introduce non-causal implementation due to overlapping time-bins. To mitigate it, we introduce chunked-FFT, a degenerate form of the STFT.

Recent research in AI has focused on using deep learning to improve the performance of adaptive filters. For example, deep learning has been used to improve the performance of adaptive filters for noise cancellation (Zhang and Wang, 2021), echo cancellation (Haubner and Kellermann, 2022), and equalization (Zhou et al., 2020). Deep learning has also been used to develop new adaptive filter architectures that are more robust to noise and interference (Alwan and Hussain, 2022). Revach et al. (2022) demonstrate how deep learning can be used to improve the performance of Kalman filtering (Kalman, 1960), a classical control algorithm.

3 Method

3.1 Overview

We start by discussing the main design choices of our architecture.

Chunking and the combination of local and global models Given the quadratic complexity of transformers, chunking is a common practice for computing short-range attention efficiently. However, despite excelling in short-range tasks, full-length transformers often struggle to handle long-range dependencies and often perform comparably to local-attention-based transformers (Xiong et al., 2021). Recent studies have demonstrated that a combination of local and global transformers can achieve state-of-the-art performance on long-range tasks (Ivgi et al., 2023; Yu et al., 2023; Hutchins et al., 2022). Inspired by these findings, we introduce local attention as the local model, which is combined with a novel type of global convolution as the global model. Furthermore, in contrast to (Hutchins et al., 2022; Bulatov et al., 2023), our global model does not use recurrent computations, since it severely restricts parallelization.

Adaptive IIR Filters In MEGA (Ma et al., 2022), it was demonstrated that incorporating an EMA at the beginning of each transformer block improves transformer performance in long-range tasks. EMA can be viewed as a convolution operation using simple first-order IIR filters. Motivated by this finding, we adopt a more versatile and expressive convolution approach that utilizes adaptive filters generated by a hypernetwork. Since the hypernetwork is an integral part of our global model, it employs global convolutions. Specifically, the regularized global convolution of (Fu et al., 2023a) is used, as the most straightforward option. A common challenge with hypernetworks is ensuring relatively small output sizes. In this regard, leveraging IIR filters, which have only a few parameters, is a reasonable choice.

3.2 The Focus Layer

In this section, we describe the focus attention head, our primary contribution. This head is integrated into the MEGA backbone, as visualized in Fig 1. Let x be the input for the focus layer, where $x \in \mathcal{R}^{L \times D}$, L is the sequence length, and D is the input's dimension. Our method, termed Focus, utilizes the foundations of adaptive filtering theory to cope with very long stochastic sequences. Given the seasonality of the sequence, the resolution of the FFT is determined by its size in each time-bin. Denote the size of the FFT in a single time-bin as $NFFT$.

The first component of the Focus layer is the hypernetwork, H. The output of H is Θ, which

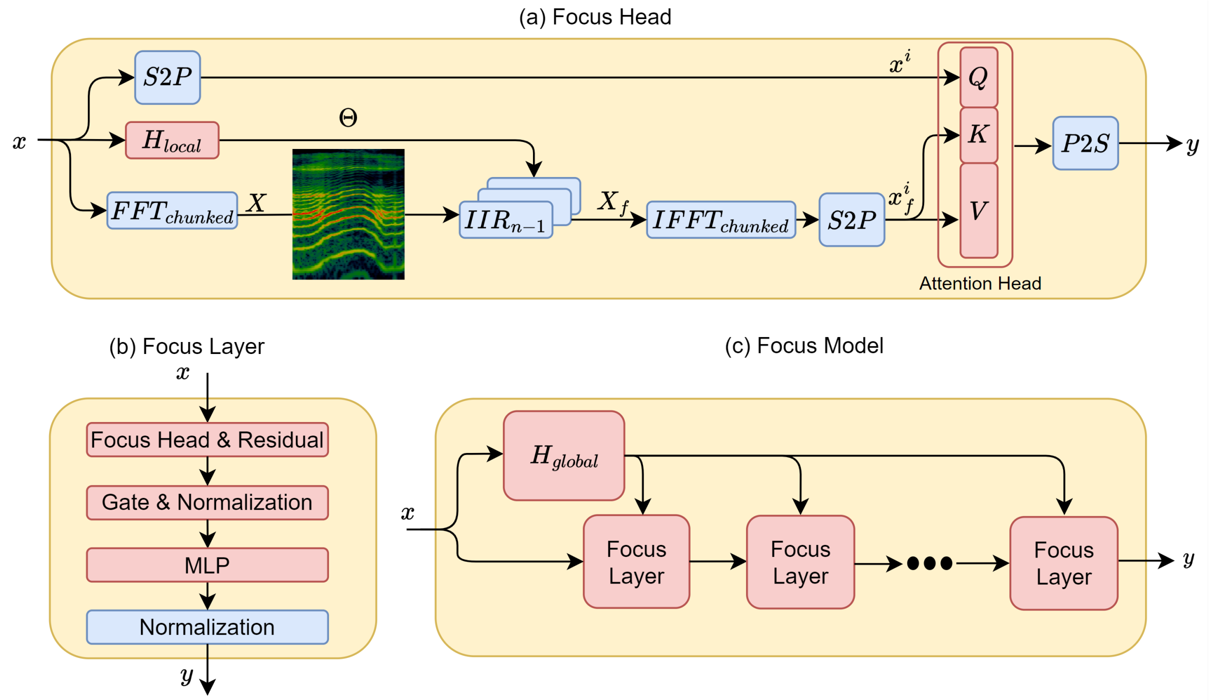

Figure 1: Focus Architecture: (a) The architecture of a single head. (b) The obtained layer. (c) The entire model. The architecture of the model and layer are defined similarly to MEGA (Ma et al., 2022). Blocks in blue are not learned, while blocks in red are learned parameters. $S2P$ (serial to parallel) and $P2S$ (parallel to serial) are the chunking and the de-chunking operations, respectively.

is the set of IIR kernels used for the forward processing of the sequence. Θ has a dimension of $Nbins \times D \times F \times 2$, denoting F kernels, each with a kernel size of two for D feature channels. The kernel is unique per time-bin, $Nbins$, which makes the filter adaptive to changes over time.

$$\Theta = H(x)\,. \tag{1}$$

H has two main components. The first is a shallow global convolution (Fu et al., 2023b) based sub-model that is followed by adaptive max pooling (over each feature channel) (Pytorch, 2023) with a size of $O \times Nbins$, where O is the oversampling factor.

$$e = \mathrm{MaxPool}(\mathrm{GlobalConv}(x)) \tag{2}$$

where e is the embedding from processing x using the global convolution layer. This computation can be shared across multiple Focus layers and can be split into local (H_{local}) and global (H_{global}) subcomponents, as in most of our experiments, thus reducing substantially the computational cost. Furthermore, the embedding is permuted such that the feature space has the size O while $Nbins$ is added to the batch dimension for parallel computing.

The second component of H is a 2-layer MLP with sigmoid activations that maps the embedding e to a tensor with size $Nbins \times D \times F \times 2$. With mapping of latent dimension O to $2 \cdot F$

$$\Theta = MLP(e)\,, \tag{3}$$

where Θ is the IIR kernel, with size $Nbins \times D \times F \times 2$. MLP is the forward MLP mapping, as described above. Since H is a hypernetwork, the initialization of the last layer of the MLPs follows (Chang et al., 2020). The rest of the layers follow the Xavier initialization (Glorot and Bengio, 2010). The input x is split into non-overlapping time bins, where each time bin is passed through the FFT of the size $NFFT$. Denote the input in the r-th time bin as x_r.

$$X[\omega, r] = FFT(x_r)\,, \tag{4}$$

where ω is the normalized frequency variable, sampled evenly on 2π range, and r is the index over the different time bins.

A note about causality To maintain a fully causal model that is applicable to auto-regressive tasks, each time bin is processed on its own and is not overlapped with other time bins. In addition, Θ is shifted right by one time bin, such that the sequence at time bin i is processed by the kernel computed from time bin $i-1$.

For each time-bin index, the corresponding IIR filter is applied. The IIR filter of order 2 has the following frequency response, denote it as IIR_{imp}

$$IIR_{imp}(f) = \frac{1}{1 + \Theta[0] \cdot e^{-j \cdot 2\pi f} + \Theta[1] \cdot e^{-j \cdot 4\pi f}}\,, \tag{5}$$

Since a Sigmoid activation is used for the last layer of H, it is guaranteed that Θ's elements are positive real numbers smaller than 1. Further analysis and reasoning behind the specific selections made are presented in Sec. 4.

Recall that in the frequency domain, the equivalent of filtering is multiplying with the conjugated impulse response,

$$X_f = X \circ IIR^*_{imp}\,, \tag{6}$$

where the conjugation is denoted by a star and $\circ$ is the elementwise (Hadamard) multiplication. The hyper-dimension F defined earlier as the filter-bank size is collapsed via regular sum operation, denote the collapsed tensor as X_c

$$X_c = \mathbf{1} \cdot X_f\,, \tag{7}$$

where $\mathbf{1}$ is an all-ones vector with size $1 \times F$. The collapsed tensor is the short-time Fourier representation of the original sequence filtered with adaptive filter kernels. The frequency representation goes through the inverse chunked Fourier transform (IFFT), to obtain the time-domain sequence.

$$x_f = IFFT(X_c) \tag{8}$$

The time-domain sequence has the same dimensions as the original sequence, yet, by applying an adaptive filter to it, we furnish it with an induction bias that helps smaller context attention head to cope with complicated tasks.

The sequence is split into C separate non-overlapping chunks. Denote the chunk length as M, such that $L = MC$. We denote the chunk of signal with uppercase i, and use square brackets for indexing, starting with index 0, as follows

$$x^i = (x[iM], x[iM{+}1], \ldots, x[(i{+}1)M{-}1]) \tag{9}$$

All chunks are processed in parallel with the same small attention head,

$$y^i = Atten(Qx^i, Kx^i_f, Vx^i_f)\,, \tag{10}$$

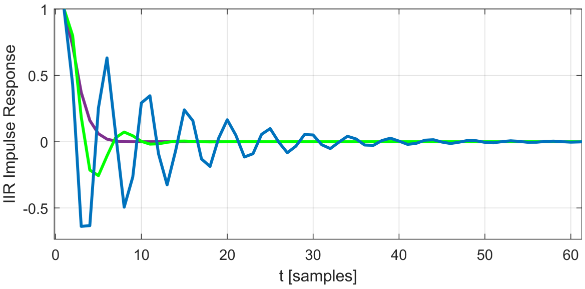

Figure 2: Filter responses for three random filters with the specific denominator structure of Eq. 16.

where Q, K, V are the query, key, and value matrices that map each chunk to their latent corresponding space. Note that the attention head is causal, as it uses lower triangle masking. The chunks, y^i, are rearranged to form a complete signal, with sequence length L, denote it as y. Following (Ma et al., 2022), the output of the Focus layer, y, passes through reset gate γ, the update gate ψ. Specifically,

$$\gamma = \text{SiLU}(x_f W_\gamma + b_\gamma) \tag{11}$$

$$\phi = \text{sigmoid}(x_f W_\phi + b_\phi) \tag{12}$$

$$z = \text{SiLU}(x_f W_h + (\gamma \circ y) U_h + b_h)\,, \tag{13}$$

where W_γ, W_ϕ and W_h are learned matrices with size $D \times D$. b_γ, b_ϕ, and b_h are learned biases with size D. SiLU stands for the sigmoid linear unit (Elfwing et al., 2018).

The final output is the gated summation of the gated attention and the input sequence,

$$o = \phi \circ z + (1 - \phi) \circ x \tag{14}$$

4 Analysis

IIR and FIR Filters IIR (Infinite Impulse Response) and FIR (Finite Impulse Response) filters are two commonly used types of digital filters with distinct characteristics. The main difference between them lies in their impulse response and filtering properties. FIR filters have a finite impulse response, meaning that the filter output is based solely on a finite number of past input samples. In contrast, IIR filters have an infinite impulse response, allowing the filter output to depend on both past and future input samples.

One advantage of IIR filters is their ability to achieve a desired frequency response with fewer coefficients compared to FIR filters. This makes IIR filters more computationally efficient, requiring fewer calculations and lower memory requirements.

Consequently, in control feedback systems where real-time operation and computational efficiency are crucial, IIR filters are often preferred.

Additionally, IIR filters can exhibit higher selectivity and sharper roll-off in the frequency domain compared to FIR filters. This characteristic can be advantageous in control feedback systems, where precise control over specific frequency components is necessary.

Stability IIR filters can be more sensitive to quantization errors and can be prone to instability if not properly designed. The presence of feedback loops in control systems can further impact stability considerations. Therefore, careful attention must be given to stability analysis and appropriate filter design techniques to ensure reliable performance.

The filter can be described in the frequency domain as a rational polynomial function of the complex exponent $e^{-j2\pi f}$. Denote the complex exponent as S.

$$S = e^{-j2\pi f} \tag{15}$$

A second-order system can be described as follows,

$$IIR(S) = \frac{1}{aS^2 + bS + 1} \tag{16}$$

Specifically solving for general b, a gives,

$$IIR(t) = \alpha e^{t(-\frac{b}{2a} - \frac{\sqrt{b2-4a)}}{2a})} + \beta e^{t(-\frac{b}{2a} + \frac{\sqrt{b2-4a)}}{2a})}, \tag{17}$$

where α and β are normalizing factors. Denote the term under the square root as discriminant, Δ. For any $b \geq 0$ and $a \geq 0$ the following holds:

$$\Delta \leq b^2 \;\; \forall \{a, b\} \geq 0\,. \tag{18}$$

In order to guarantee stability, both exponents should decay, which leads to the requirement that the real part must be negative.

$$Re\{-\frac{b}{2a} \pm \frac{\sqrt{\Delta)}}{2a})\} \leq 0\,. \tag{19}$$

This can be achieved if b is positive. In the scenario where $\Delta \leq 0$, the exponents have an imaginary part, causing it to oscillate. This is called an under-damped response. This response is stable, yet more expressive than Exponential Moving Average (EMA) filters, as found in MEGA (Ma et al., 2022). The frequency of the sine, ω, in this case is

$$\omega = \frac{\sqrt{\Delta}}{2a}, \tag{20}$$

and the time domain impulse response reads,

$$IIR(t) = \gamma e^{-t\frac{b}{2a}} sin(\omega t + \phi)\,, \tag{21}$$

where γ is the normalizing factor, and ϕ is the phase from aggregating both sine and cosine functions with the same frequency. Note that for orders above 2, there is no simple condition that guarantees that the real part will be negative and the response stable.

To demonstrate the oscillating behavior of the generated IIR filters, the time-domain impulse responses of some random kernels are drawn in Fig. 2. In this plot, the purple kernel acts as EMA, while other kernels have a more complicated response. However, all filters decay as time increases, which leads to a stable response and has been identified by Li et al. (2022) as an essential property for capturing long-range dependencies.

Time Complexity The global conv time complexity is

$$\text{GlobalConv} \approx O(Llog(L) \cdot D) \tag{22}$$

This is due to the FFT and IFFT that the signal is passed through. This computation is done only once and is shared through multiple layers of Focus. The MLP is mapping between D and F.

$$MLP \approx O(DF) \tag{23}$$

The chunked-FFT complexity depends on the size of the FFT used ($NFFT$). Denote the size of a single time bin as R,

$$R = \frac{L}{Nbins} \tag{24}$$

The time complexity of a single time bin is $O(Rlog(R))$. The total time complexity of the chunked FFT reads,

$$\text{FFT}_{\text{chunked}} \approx O(Llog(R)) \tag{25}$$

Next, the time complexity of the attention head depends on the size of the context length M,

$$\text{Atten} \approx O(CM^2D + CMD^2) \tag{26}$$

The total time complexity of the Focus layer is, therefore,

$$\text{Focus} \approx O(Llog(L) \cdot D + CM^2D) \tag{27}$$

where we neglected smaller terms when the sequence length is large (greater than dimensions).

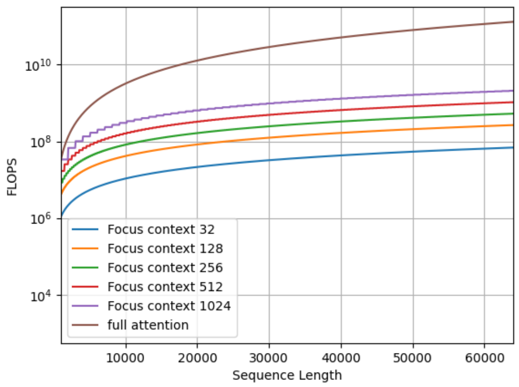

Figure 3: Time Complexity of the Focus layer and of Attention, increasing sequence length

Recalling that $L = MC$, and rearranging terms, we have,

$$\text{Focus} \approx O(DL \cdot (log(L) + M)) \tag{28}$$

obtaining sub-quadratic time complexity with respect to input sequence length. A visual comparison of overall complexity versus the standard attention head is depicted in Fig. 3.

Expressiveness An emerging class of diagonal linear RNNs (Orvieto et al., 2023; Gupta et al., 2022b) recently achieved near SOTA results in several long-range tasks. They include complex and real variants, as well as diagonal state-space layers (Gupta et al., 2022a; Gu et al., 2022). The following recurrent rule describes each channel of those layers:

$$s[t] = As[t-1] + Bx[t], \quad y[t] = Cs[t] + Dx[t] \tag{29}$$

where $s[t]$ is the recurrent state at time t. By isolating $s[t-1]$, we can rewrite Eq. 29 as follows:

$$s[t-1] = \frac{1}{C}y[t-1] - \frac{D}{C}x[t-1] \tag{30}$$

$$y[t] = CAs[t-1] + (CB+D)x[t] = \tag{31}$$

$$Ay[t-1] + (CB+D)x[t] + ADx[t-1]$$

Recall that the differential equation of an IIR filter of order 2 can be represented as follows:

$$y[t] = b_0x[t] + b_1x[t-1] + b_2x[t-2] - \tag{32}$$

$$a_1y[t-1] - a_2y[t-2]$$

By substituting the values of $b_0 = CB+D$, $b_1 = AD$, $a_1 = -A$, $b_2 = a_2 = 0$, it becomes evident that the IIR filter can be constrained to a linear SSM. In machine learning, D is often omitted in SSMs or diagonal RNNs, since it can be seen as a parameter-based skip-connection. In this case, the SSM can be represented by an IIR filter of order 1.

As mentioned earlier, higher-order filters can introduce stability issues. Therefore, our decision to utilize IIR filters of order 2 is justified, as we opt for the most expressive IIR filters that still maintain stability during training.

5 Experiments

Below we present experimental results for the proposed Focus layer. In addition to our full method, we introduce an ablation to evaluate the importance of adaptive filtering, in which instead of the hypernetwork H, the IIR filters are conventional learned parameters. This ablation is denoted by "Focus-H".

5.1 In-context learning

In order to evaluate our method relative to other state-of-the-art long-range architectures, such as (Poli et al., 2023), (Dai et al., 2019), the associative recall synthetic task is evaluated. The associative recall task was first introduced in (Elhage et al., 2021) and is part of a number of simple yet informative tasks that test the capabilities of the model in processing long-range sequences.

In the associative recall task, each string is formed by concatenating key-value pairs sampled randomly from a dictionary. The model should output the correct value given a singular key, regardless of whether the key is in the long sequence.

Similarly to Poli et al. (2023), we employ the associative recall task in order to explore the memory capabilities of our model.

In all synthetic data experiments the same shared hyperparameters are used, with the exception of the sequence length. The hyperparameters are depicted in in Appendix A. The AdamW optimizer (Loshchilov and Hutter, 2018) is used.

As can be seen in Tab. 1, our model is able to obtain an accuracy of 100% for all sequence lengths, without overfitting, despite the low number of examples (2000), and with no memory explosion thanks to linear scaling with input size. These results show that the Focus mechanism is able to improve the performance of regular transformers to be on par with Heyna (Poli et al., 2023), with smaller footprint. In addition, the ablation experiment shows the importance of adaptive filtering,

Table 1: Test accuracy (%) for associative recall on long sequences of length L and a vocabulary size of 30. NF - not feasible to test. NR = not reported.

L	Focus	Focus-H	Hyena	FlashTransformer	Transformer
30	100.0	100.0	100.0	100.0	100.0
1K	100.0	98.0	100.0	95.0	100.0
8K	100.0	85.3	100.0	NR	NF
32K	100.0	34.6	100.0	32.4	NF
64K	100.0	28.0	100.0	26.7	NF

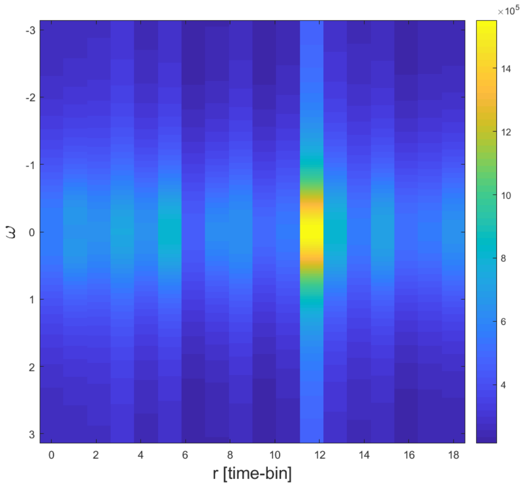

Figure 4: Frequency Response of IIR kernels, for 1K sequence split into 18 time bins. The important key is found in the 12th time bin.

i.e. estimating the filter kernels online to focus the attention mechanism on important sub-sequences where the results degrade in the ablation due to the greater sequence length.

The associative recall task is used not only for benchmarking, but also to gain insights into the adaptive filtering mechanism. As can be seen in Fig. 4, the frequency response of the adaptive filtering is plotted for this task. In this specific run, the important key is found in the 12th time bin. The frequency response of the IIR filters is almost 5 orders of magnitude higher than for nearby time bins. This effect demonstrates the “Focus” mechanism. Note that using only 2 parameters for the IIR kernel, the filters are able to differentiate between important and unimportant time bins with 5 orders of magnitude. This supports our design choice of IIR filter, seeing that with a kernel size as low as 2 the filter is still sharp enough.

5.2 Language Modeling

The enwiki8 dataset is a byte-level dataset consisting of the first 100 million bytes of a Wikipedia XML dump. It is a commonly used dataset for benchmarking character-level language models.

The Text8 dataset is a corpus of text used for training and evaluating language models. It is a subset of the Wikipedia dump from March 2006 and consists of 90 million characters. The text is tokenized and lowercased, and each token is assigned a unique id.

The metric used to evaluate language models on enwiki8 and Text8 is bits per character (BPC). The lower the BPC, the better the language model. To compute the BPC the average cross-entropy is computed in the log2 basis

$$BPC = \frac{1}{L} Log_2 \, \mathrm{CrossEntropy}(P, \hat{P}) \,, \qquad (33)$$

where P is the target distribution, and $\hat{P}$ is the output distribution. L is the sequence length. To maintain the same capabilities such as Mega (Ma et al., 2022), we used 8 layers of the Focus layer, with a hidden dimension of 1024 and an input dimension of 512.

The enwiki8 results are reported in Tab. 2. Evidently, Focus outperforms both Mega (Ma et al., 2022) and Transformer XL (Dai et al., 2019), despite a much lower number of parameters, performing on-par with GPT2(Radford et al., 2019) (zero-shot) but with a fraction of its parameters and substantially less FLOPS. The same occurs for the Text8 dataset, reported in Tab. 3. While the ablation is inferior to Transformer-XL, with the full method Focus is on par with GPT2.

5.3 1-D image classification

We evaluated our model on the sequential MNIST task, a challenging problem that requires models

Table 2: BPC for enwiki8 dataset.

Model	#params	BPC
Transformer XL (Dai et al., 2019)	277M	0.99
Mega (Ma et al., 2022)	39M	1.02
GPT2 (Radford et al., 2019)	1542M	0.94
Focus-H (ablation)	21M	1.06
Focus	22M	**0.94**

Table 3: BPC for text8 dataset.

Model	#params	BPC
Transformer XL (Dai et al., 2019)	277M	1.08
GPT2 (Radford et al., 2019)	1542M	0.98
Focus-H (ablation)	21M	1.10
Focus	22M	**0.98**

Table 4: Accuracy for sMNIST, pMNIST

Model	sMNIST	pMNIST
Transformer	98.9	97.9
S4 (Gu et al., 2021a)	99.6	98.7
Focus-H (ablation)	98.9	98.0
Focus	**99.7**	**98.8**

to capture long-range dependencies. Permuted MNIST is a variant of MNIST where the order of pixels in each image is scrambled, making the task more challenging. Following S4, we use a hidden dimension of 512 but to save resources, we use 6 layers and not more. The results are listed in Tab. 4. Focus has an accuracy of 99.7% (98.8%) on unpermuted (permuted) MNIST, outperforming the transformer and S4.

5.4 Efficiency Comparison

To assess the efficiency of the Focus layer, we measure peak memory usage and inference speed. We compare several related models on the association recall task. This task involves processing a sequence of 1K tokens, which represents the maximal fit in memory for transformers. The results are presented in Tab. 5. As can be seen, Focus exhibits the lowest peak memory consumption, using only 38% of the memory consumed by the Transformer model. However, its inference speed is slightly slower than the other methods.

Model	Inference time	Memory
Transformer	x1	x1
S4	x1.58	x0.43
MEGA	x1.49	x0.57
Focus	x1.75	x0.38

Table 5: Comparison of inference speed and peak memory consumption for related models.

6 Conclusions

Attention models are extremely powerful for modeling sequences, as demonstrated by the seminal work of Vaswani et al. (2017). Indeed, transformers have revolutionized the way deep learning is practiced, leading to unprecedented performance across almost all studied AI domains.

However, transformers have quadratic complexity in the sequence length, which can impact their efficiency, and often struggle to perform optimally in tasks that involve long-range dependencies (Tay et al., 2020). In this work, we present a dynamic filtering approach that enables us to subsequently employ attention within much shallower architectures. As our ablation shows, the dynamic nature of these filters is crucial to the success of the layer. Similarly crucial is the use of IIR filters, and we analyze the regime in which these are stable.

7 Limitations

While this paper presents promising results, there are a few limitations to consider. Firstly, although we are the first to utilize IIR filters for long-range tasks, we have not examined sequence models solely based on IIR filters. Additionally, we have not investigated the impact of different types of hyper-global convolution on performance. Furthermore, IIR filters can be computed using recurrent rules or convolution. While the convolutional view is more natural for training, the recurrent view presented in Eq. 32, can be leveraged for efficient auto-regressive generation. This can lead to a significant reduction in the time and space complexity of the layer during inference, which is beneficial for real-time applications.

8 Acknowledgments

This work was supported by a grant from the Tel Aviv University Center for AI and Data Science (TAD). It is part of a PhD research conducted by the first author.

References

Arij Al Adel. 2022. Global memory transformer for processing long documents. In *Advances in Neural Computation, Machine Learning, and Cognitive Research VI: Selected Papers from the XXIV International Conference on Neuroinformatics, October 17-21, 2022, Moscow, Russia*, pages 343–352. Springer.

Arij Al Adel and Mikhail S Burtsev. 2021. Memory transformer with hierarchical attention for long document processing. In *2021 International Conference Engineering and Telecommunication (En&T)*, pages 1–7. IEEE.

Nuha A. S. Alwan and Zahir M. Hussain. 2022. Deep learning for robust adaptive inverse control of nonlinear dynamic systems: Improved settling time with an autoencoder. *Sensors*, 22(16).

Luca Bertinetto, João F Henriques, Jack Valmadre, Philip Torr, and Andrea Vedaldi. 2016. Learning feed-forward one-shot learners. In *Advances in Neural Information Processing Systems*, pages 523–531.

Aydar Bulatov, Yuri Kuratov, and Mikhail S Burtsev. 2023. Scaling transformer to 1m tokens and beyond with rmt. *arXiv preprint arXiv:2304.11062*.

Oscar Chang, Lampros Flokas, and Hod Lipson. 2020. Principled weight initialization for hypernetworks. In *Int. Conf. on Learning Representations*.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. 2020. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022a. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359.

Tri Dao, Daniel Y Fu, Khaled K Saab, Armin W Thomas, Atri Rudra, and Christopher Ré. 2022b. Hungry hungry hippos: Towards language modeling with state space models. *arXiv preprint arXiv:2212.14052*.

Shmuel Bar David, Itamar Zimerman, Eliya Nachmani, and Lior Wolf. 2023. Decision s4: Efficient sequence-based rl via state spaces layers. In *The Eleventh International Conference on Learning Representations*.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11.

N Elhage, N Nanda, C Olsson, T Henighan, N Joseph, B Mann, A Askell, Y Bai, A Chen, T Conerly, et al. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*.

Quentin Fournier, Gaétan Marceau Caron, and Daniel Aloise. 2021. A practical survey on faster and lighter transformers. *ACM Computing Surveys*.

Daniel Y Fu, Elliot L Epstein, Eric Nguyen, Armin W Thomas, Michael Zhang, Tri Dao, Atri Rudra, and Christopher Ré. 2023a. Simple hardware-efficient long convolutions for sequence modeling. *arXiv preprint arXiv:2302.06646*.

Daniel Y Fu, Elliot L Epstein, Eric Nguyen, Armin W Thomas, Michael Zhang, Tri Dao, Atri Rudra, and Christopher Re. 2023b. Simple hardware-efficient long convolutions for sequence modeling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10373–10391. PMLR.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

Karan Goel, Albert Gu, Chris Donahue, and Christopher Ré. 2022. It's raw! audio generation with state-space models. In *International Conference on Machine Learning*, pages 7616–7633. PMLR.

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

Albert Gu, Karan Goel, and Christopher Ré. 2021a. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*.

Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. 2021b. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585.

Ankit Gupta and Jonathan Berant. 2020. Gmat: Global memory augmentation for transformers. *arXiv preprint arXiv:2006.03274*.

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

Ankit Gupta, Harsh Mehta, and Jonathan Berant. 2022b. Simplifying and understanding state space models with diagonal linear rnns. *arXiv preprint arXiv:2212.00768*.

David Ha, Andrew Dai, and Quoc V Le. 2016. Hypernetworks. *arXiv preprint arXiv:1609.09106*.

Thomas Haubner and Walter Kellermann. 2022. Deep learning-based joint control of acoustic echo cancellation, beamforming and postfiltering.

Yun He, Steven Zheng, Yi Tay, Jai Gupta, Yu Du, Vamsi Aribandi, Zhe Zhao, YaGuang Li, Zhao Chen, Donald Metzler, et al. 2022. Hyperprompt: Prompt-based task-conditioning of transformers. In *International Conference on Machine Learning*, pages 8678–8690. PMLR.

DeLesley Hutchins, Imanol Schlag, Yuhuai Wu, Ethan Dyer, and Behnam Neyshabur. 2022. Block-recurrent transformers. *arXiv preprint arXiv:2203.07852*.

Md Mohaiminul Islam, Mahmudul Hasan, Kishan Shamsundar Athrey, Tony Braskich, and Gedas Bertasius. 2022. Efficient movie scene detection using state-space transformers. *arXiv preprint arXiv:2212.14427*.

Maor Ivgi, Uri Shaham, and Jonathan Berant. 2023. Efficient long-text understanding with short-text models. *Transactions of the Association for Computational Linguistics*, 11:284–299.

Rudolph Emil Kalman. 1960. A new approach to linear filtering and prediction problems.

Boris Kuznetsov, Julian D Parker, and Fabián Esqueda. 2020. Differentiable iir filters for machine learning applications. In *Proc. Int. Conf. Digital Audio Effects (eDAFx-20)*, pages 297–303.

Yuhong Li, Tianle Cai, Yi Zhang, Deming Chen, and Debadeepta Dey. 2022. What makes convolutional models great on long sequence modeling? *arXiv preprint arXiv:2210.09298*.

Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam.

Shahar Shlomo Lutati and Lior Wolf. 2023. Ocd: Learning to overfit with conditional diffusion models. In *ICML*.

Xuezhe Ma, Chunting Zhou, Xiang Kong, Junxian He, Liangke Gui, Graham Neubig, Jonathan May, and Luke Zettlemoyer. 2022. Mega: moving average equipped gated attention. *arXiv preprint arXiv:2209.10655*.

Harsh Mehta, Ankit Gupta, Ashok Cutkosky, and Behnam Neyshabur. 2022. Long range language modeling via gated state spaces. *arXiv preprint arXiv:2206.13947*.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. 2023. Resurrecting recurrent neural networks for long sequences. *arXiv preprint arXiv:2303.06349*.

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. *arXiv preprint arXiv:2302.10866*.

Pytorch. 2023. Adaptivemaxpool2d¶.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Guy Revach, Nir Shlezinger, Xiaoyong Ni, Adria Lopez Escoriza, Ruud J. G. van Sloun, and Yonina C. Eldar. 2022. KalmanNet: Neural network aided kalman filtering for partially known dynamics. *IEEE Transactions on Signal Processing*, 70:1532–1547.

David W Romero, Anna Kuzina, Erik J Bekkers, Jakub M Tomczak, and Mark Hoogendoorn. 2021. Ckconv: Continuous kernel convolution for sequential data. *arXiv preprint arXiv:2102.02611*.

George Saon, Ankit Gupta, and Xiaodong Cui. 2023. Diagonal state space augmented transformers for speech recognition. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Filip Szatkowski, Karol J Piczak, Przemysław Spurek, Jacek Tabor, and Tomasz Trzciński. 2022. Hypersound: Generating implicit neural representations of audio signals with hypernetworks. *arXiv preprint arXiv:2211.01839*.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2020. Long range arena: A benchmark for efficient transformers. *arXiv preprint arXiv:2011.04006*.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–28.

Ali Vardasbi, Telmo Pires, Robin M. Schmidt, and Stephan Peitz. 2023. State spaces aren't enough: Machine translation needs attention. *ArXiv*, abs/2304.12776.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Johannes Von Oswald, Christian Henning, Benjamin F Grewe, and João Sacramento. 2019. Continual learning with hypernetworks. *arXiv preprint arXiv:1906.00695*.

Jue Wang, Wentao Zhu, Pichao Wang, Xiang Yu, Linda Liu, Mohamed Omar, and Raffay Hamid. 2023. Selective structured state-spaces for long-form video understanding. *arXiv preprint arXiv:2303.14526*.

Junxiong Wang, Jing Nathan Yan, Albert Gu, and Alexander M Rush. 2022. Pretraining without attention. *arXiv preprint arXiv:2212.10544*.

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*.

Genta Indra Winata, Samuel Cahyawijaya, Zhaojiang Lin, Zihan Liu, and Pascale Fung. 2020. Lightweight and efficient end-to-end speech recognition using low-rank transformer. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6144–6148. IEEE.

Wenhan Xiong, Barlas Oğuz, Anchit Gupta, Xilun Chen, Diana Liskovich, Omer Levy, Wen-tau Yih, and Yashar Mehdad. 2021. Simple local attentions remain competitive for long-context tasks. *arXiv preprint arXiv:2112.07210*.

Lili Yu, Dániel Simig, Colin Flaherty, Armen Aghajanyan, Luke Zettlemoyer, and Mike Lewis. 2023. Megabyte: Predicting million-byte sequences with multiscale transformers. *arXiv preprint arXiv:2305.07185*.

Chris Zhang, Mengye Ren, and Raquel Urtasun. 2019. Graph hypernetworks for neural architecture search. In *7th International Conference on Learning Representations, ICLR 2019*.

Hao Zhang and DeLiang Wang. 2021. Deep anc: A deep learning approach to active noise control. *Neural Networks*, 141:1–10.

Michael Zhang, Khaled K Saab, Michael Poli, Tri Dao, Karan Goel, and Christopher Ré. 2023. Effectively modeling time series with simple discrete state spaces. *arXiv preprint arXiv:2303.09489*.

Qingyi Zhou, Fan Zhang, and Chuanchuan Yang. 2020. AdaNN: Adaptive neural network-based equalizer via online semi-supervised learning. *Journal of Lightwave Technology*, 38(16):4315–4324.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, pages 27268–27286. PMLR.

Simiao Zuo, Xiaodong Liu, Jian Jiao, Denis Charles, Eren Manavoglu, Tuo Zhao, and Jianfeng Gao. 2022. Efficient long sequence modeling via state space augmented transformer. *arXiv preprint arXiv:2212.08136*.

A Hyperparameters

The hyperparameters for the associative recall task are provided in Tab. 6. Hyperparameters that differ in other experiments are reported in the respective sections. For example, in Sec. 5.2, eight layers of the Focus layer are used, with a hidden dimension of 1024 and an input dimension of 512, in order to compare with Mega on similar terms.

Table 6: Hyperparameter settings for the synthetic associative recall task

Optimizer	AdamW
Optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.98$
Vocabulary Size	30
NFFT	$L/4$
F	1
C	32
Learning Rate	1E-4
Batch Size	32
Num Samples	2000
Warmup epochs	10
Number of Layers	2
Width	64