Multi-Head State Space Model for Speech Recognition

Yassir Fathullah[1,2]*, *Chunyang Wu*[2], *Yuan Shangguan*[2], *Junteng Jia*[2], *Wenhan Xiong*[2],
Jay Mahadeokar[2], *Chunxi Liu*[2], *Yangyang Shi*[2], *Mark J. F. Gales*[1], *Ozlem Kalinli*[2], *Mike Seltzer*[2]

[1]University of Cambridge, UK
[2]Meta AI, USA

yf286@cam.ac.uk, chunyang@meta.com

Abstract

State space models (SSMs) have recently shown promising results on small-scale sequence and language modelling tasks, rivalling and outperforming many attention-based approaches. In this paper, we propose a multi-head state space (MH-SSM) architecture equipped with special gating mechanisms, where parallel heads are taught to learn local and global temporal dynamics on sequence data. As a drop-in replacement for multi-head attention in transformer encoders, this new model significantly outperforms the transformer transducer on the LibriSpeech speech recognition corpus. Furthermore, we augment the transformer block with MH-SSMs layers, referred to as the Stateformer, achieving state-of-the-art performance on the LibriSpeech task, with word error rates of 1.76%/4.37% on the development and 1.91%/4.36% on the test sets without using an external language model.

Index Terms: speech recognition, transducer, librispeech, state space model, attention-free, transformer, stateformer

1. Introduction

Recurrent neural networks (RNNs) have historically been a core approach for a wide range of sequence modelling tasks such as speech recognition [1, 2], machine translation [3, 4] and language modelling [5, 6]. However, RNNs were rapidly replaced with the introduction of the transformer [7] and large pre-trained models [8]. The effectiveness of the transformer has also caused other fields such as computer vision to consolidate towards attention-based models [9, 10].

One of the key reasons behind the success of transformers and their widespread use is the self-attention mechanism. Unlike previous approaches, self-attention was shown to be highly effective at capturing global features of a sequence by modelling all pairwise interactions. Furthermore, while transformers are exceptionally good at capturing global long-range dependencies they are less able in modelling local patterns. To this end, there has been a range of work on combining convolutional networks with attention for sequence modelling, and has been found to be highly effective for speech recognition [11, 12, 13].

Meanwhile, the deep learning community has slowly been paying more attention to alternative recurrent neural network approaches to effectively and efficiently model sequences [14, 15, 16, 17]. Specifically, a well-established signal processing and control theory technique, the state space model (SSM) [18], which historically has been widely used in many continuous or discrete time-series and control problems [19, 20] has been under renewed scrutiny. However, the recurrent time-variant nature of general SSMs has traditionally made them computationally expensive and inaccessible to many large-scale sequence tasks. Nonetheless, recent work has shown that it is possible to simplify and scale state space models and train them in parallelized manner, by equivalently rephrasing them as a convolution with variable-length kernels [14, 15]. Further work has also shown that simplified, structured versions of the time-invariant state space model can be highly efficient, handle long-range dependencies and be a formidable alternative to self-attention on some sequence tasks, such as language modelling [21, 22, 23].

In this work, we evaluate and extend the work on SSMs for speech recognition, proposing a multi-head state space model, equipped with a novel gating mechanism. Since SSMs have shown promising performance on long sequence tasks we hypothesize that a multi-headed approach could better handle both short and long-term modelling simultaneously. We investigate the use of such multi-head state space models both as a replacement and complement to self-attention in the acoustic encoder of a neural network transducer model. Our technical contributions include:

(a) *Stacked and multi-head generalization.* We extend the SSM approach by allowing multi-head processing of linearly projected lower-dimensional signals and stack such a layer for better performance.

(b) *Head gating.* We propose an inter-head gating approach in which different SSMs within the multi-head layer communicate by gating each other.

(c) *Combination with attention.* We also augment the transformer encoder by including a bidirectional SSM residual block prior to the attention block for state-of-the-art performance. This model is referred to as the *Stateformer*.

With these contributions, we advance the state of attention-free models on the LibriSpeech speech recognition task, outperforming strong attention-based baselines. We also show that the Stateformer can achieve state-of-the-art performance on this task with word error rates of 1.76%/4.37% on the development and 1.91%/4.36% on the test sets, without using an external language model.

2. State Space Model: The Linear RNN

The time-invariant state space model [24] is a fully linear recurrent network taking the following form:

$$\left.\begin{aligned} \boldsymbol{x}_k &= \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{B}\boldsymbol{u}_k \\ \boldsymbol{y}_k &= \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{D}\boldsymbol{u}_k \end{aligned}\right\} \; \boldsymbol{y} = \texttt{SSM}(\boldsymbol{u}) \tag{1}$$

It simply transforms an arbitrary input signal $\boldsymbol{u}$ into an output signal $\boldsymbol{y}$ through some hidden process $\boldsymbol{x}$. Since this model is linear, it can also be phrased as a convolution [14, 25] allowing it to be trained in a parallelizable manner without recur-

*Work done during internship at Meta AI.

rences. More importantly, this model can be made highly efficient and effective by restricting the parameters $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}$ to be block-diagonal and ensuring that the transition matrix A is stable i.e., the SSM generates bounded outputs for bounded inputs [22, 26]. Furthermore, the effectiveness of this model is highly dependent on its initialization. Work has found that this system can encode the history of an input signal effectively with a proper initialization scheme [21, 27]. The combination of these ideas culminates in a model called S4 [22].

The S4 model is inherently unidirectional. For non-causal applications such as the audio-encoder for offline speech recognition, a bidirectional S4 can be used with non-linear activations and pointwise linear layers [28]:

$$\begin{aligned} \boldsymbol{y} &\leftarrow \texttt{Cat}([\texttt{S4}(\boldsymbol{u}), \texttt{Rev}(\texttt{S4}(\texttt{Rev}(\boldsymbol{u})))]) \\ \boldsymbol{y} &\leftarrow \texttt{Linear}(\texttt{Activation}(\boldsymbol{y})) \end{aligned} \tag{2}$$

The next section will build upon this model by using multiple parallel heads and introducing an inter-head gating mechanism.

3. Multi-Head State Space Model

This section will describe a number of technical and architectural proposals for the audio-encoder in the transducer. Section 3.1 introduces the stacked MH-SSM. Section 3.2 describes a novel gating mechanism which allows different SSM heads to communicate. Section 3.4 combines the MH-SSM with self-attention for a new transformer architecture. Finally, Section 3.3 describes how the MH-SSM can be used to replace the convolutional frontend.

3.1. Stacked & Multi-Head Extension

We extend the S4 layer with a significantly more flexible approach. Taking inspiration from multi-head self-attention, we project the input D_i-dimensional signal into $H \in \{2, 4, 8, \dots\}$ separate signals of dimension $\bar{D}_i = D_i/H$, and process each using an independent SSM that is randomly initialized. While this can be followed by a simple non-linear activation such as ReLU or GELU, we opt for a novel gating mechanism described in the following section. Finally, we repeat this procedure once again for a stacked module, see Figure 1. This module would then operate as a drop-in replacement for the $S4$ in Equation 2 to form a bidirectional model. This multi-head design provides the flexibility to learn both meaningful time-steps and different types of temporal dynamics on sequence data.

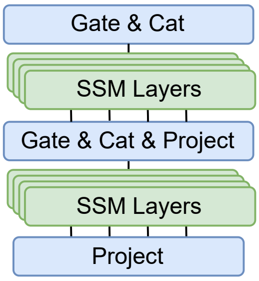

Figure 1: *The MH-SSM block first projects an input signal to several lower-dimensional signals, each fed to a separate SSM. The output is gated, concatenated and repeated a second time.*

3.2. Inter-Head Gating

By default, prior work has used the GELU activation function, see Equation 2. In some experiments [22, 28], the GLU activation was also found beneficial. However, a multi-head state space architecture with H heads offers many more gating possibilities. We propose an *inter-head gating* (IHG) approach in which half the number of heads are used to gate the remaining heads, allowing different heads to communicate and generally leading to improved results, as illustrated in Figure 2. The IHG output is computed by mixing the heads according to (where σ refers to the sigmoid):

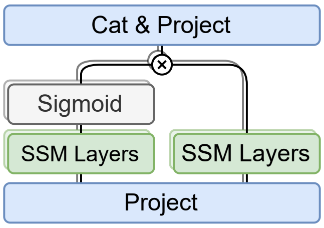

Figure 2: *Previous approaches would either apply GELU or GLU to the output of a single SSM. IHG gates the output of an SSM using another independent SSM.*

$$a^{(h)} = y^{(h)} \cdot \sigma\left(y^{(h+H/2)}\right), \quad h = \{1, ..., H/2\} \tag{3}$$

allowing different heads to communicate and gate each other, generally leading to improved results. It should be noted that the number of heads H needs to be even. While our approach is wildly different, the notion of allowing heads to communicate has also been suggested in [29] regarding attention, where linear layers mix information across heads and have been shown to improve performance.

3.3. Multi-Scale Frontend

Audio-encoders typically subsample the input sequence using a convolutional frontend to reduce sequence length and increase computational tractability [1, 30]. In this work, we utilize a multi-scale (MS) state space front end using the MH-SSM to make use of its ability to model longer-range dependencies, see Figure 3. The frontend intertwines MH-SSM blocks that capture temporal dependencies and time reduction layers which reduce the frame rate resulting in the same output size and striding as typical convolutional frontends.

3.4. Stateformer: State Space Augmented Transformer

A pure multi-head state space model architecture is attractive due to its ability to capture both short and long-range dependencies. However, since the state space model is equivalent to a linear RNN, it is expressively more limited in the temporal dimension. Therefore, we propose a model combining the MH-SSM with attention by simply inserting a pre-norm bidirectional block prior to the self-attention unit in the transformer architecture, referred to as the *Stateformer*, see Figure 4.

4. Experimental Evaluation

4.1. Data

We evaluate the proposed models on the LibriSpeech dataset [31] consisting of about 960 hours of speech data sampled at

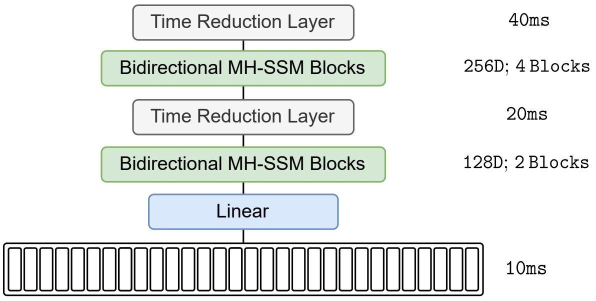

Figure 3: *Filterbanks are passed on to a linear layer with 128D outputs. These are fed through 2 smaller MH-SSM residual blocks followed by a time reduction (TR) layer which splices every two frames. This is again followed by another set of MH-SSM blocks and a TR layer resulting in 512D features with an effective 40ms frame rate.*

16kHz. SpecAugment [32] and speed perturbation [33] were used for data augmentation. We used a sliding window of 25ms with a 10ms frame shift to extract 80-dimensional filterbank.

4.2. Baselines

All speech recognition models were based on the transducer framework [30], which has three components: encoder, predictor and joiner. We train various transducers by keeping the predictor and joiner fixed, and compare various Transformer [34], Conformer [11] and S4 [22] encoders against the proposed approaches. The 80-dim input feature was first linearly projected to a dimension of 128. Furthermore, we explore the style of subsampling frontends by comparing:

(a) *Time Reduction Layers* (TR): Splicing 128-dim frames to 512 dimensions, reducing the sequence length by 4x.

(b) *Convolutional Layers* (CNN): 2D convolutional network with a total stride of 4 and 512 output channels [35, 11].

(c) *Multi-scale Layers* (MS): Proposed multi-scale frontend which intertwines MH-SSM blocks with TR layers, reducing the sequence length by 4x.

4.3. Implementation Details

Baseline and proposed models are implemented using an extension of the Fairseq framework [36]. The encoder model dimension was set to 512 and kept fixed for all experiments; the model size was controlled by the number of encoder layers. The baselines used the convolutional frontend and consists of 20-36 S4, Transformer or Conformer blocks with 8 self-attention heads (not applicable to S4), and a feed-forward net dimension of 2048. Different to [11], our Conformer baseline did not use a macaron style block as it was not found useful and had the convolutional module prior to the attention with a kernel size of 31. The prediction network consisted of a three-layer 512-dimensional LSTM with layernorm and dropout. Both the encoder and predictor outputs were projected to 1024 dimensions before being fed into an additive joiner with a single linear layer of $|\mathcal{Y}| = 4097$ sentence-piece [37] output units. Similar to the S4 baseline, our proposed MH-SSM simply replaced the self-attention layer of the transformer; the configuration of this layer, such as the number of stacks, heads, use of inter-head gating and frontend subsampler are investigated. The Stateformer uses the same setup as the transformer combined with the best MH-SSM configuration. Large (greater than 100M parameters) baselines and proposed models were also trained using auxiliary classifiers similar to [38], in which intermediate encoder outputs are trained to predict frame labels every 4 layers.

All models used the Adam optimizer [39] with a learning rate linearly warming up to the peak value in 10k iterations, fixed until the 60th epoch and thereafter, exponentially decayed by a factor of 0.96 each epoch. A dropout value of 0.10 is used in all encoders and 0.30 in all predictors and the batch size is set based on occupying maximal GPU memory. All models were trained up to 200 epochs using 32 NVIDIA A100 GPUs. Hyperparameters and level of checkpoint averaging were based on the development set.

5. Results & Discussion

5.1. Main Results

Table 1 shows the word error rate performance of our models of the large configuration on LibriSpeech with our baselines and state-of-the-art models including ContextNet [40], Transformer [34], Conformer [11, 41] and the recently introduced Branchformer [12] and E-Branchformer [13]. No external language model was used.

Table 1: *WER% performance of baseline and proposed models on Librispeech compared with best results found in the literature (no external language model). At approximately 140.3M parameters, our attention-free MH-SSM model is competitive with ContextNet and outperforms many other reported models. At 139.8M parameters, our Stateformer is competitive with the best-reported Conformer and outperforms all other models.*

Model	Params	dev		test	
		clean	other	clean	other
AED					
Branchformer [12]	116.2M	2.2	5.5	2.4	5.5
E-Branchformer [13]	148.9M	–	–	2.14	4.55
Conformer [13]	147.8M	–	–	2.16	4.74
Transducer					
Transformer [34]	139M	–	–	2.4	5.6
Transformer [42]	160M	–	–	2.2	4.7
ContextNet [40]	112.7M	2.0	4.6	**2.1**	4.6
Conformer [11]	118.8M	**1.9**	**4.4**	**2.1**	**4.3**
Conformer [41]	≃120M	2.0	4.7	2.2	4.8
Baselines					
S4 36L	129.6M	2.21	5.63	2.41	5.68
Transformer 36L	129.0M	2.16	5.28	2.32	5.34
Conformer 24L	133.7M	1.95	4.84	2.21	5.04
Proposed Models					
MH-SSM 32L	140.3M	1.80	4.96	2.01	4.61
Stateformer 25L	139.8M	**1.76**	**4.37**	**1.91**	**4.36**

The performance of our (attention-free) multi-head state space model (MH-SSM) is able to achieve competitive results of 1.80/4.96/2.01/4.61 outperforming existing transformer transducers and competing with ContextNet. It also significantly outperforms the original S4 which is unable to outperform transformer baselines. Furthermore, this model is able to outperform one of the Conformer transducers [41] on all but dev-other. The Stateformer further pushes the performance by combining MH-SSM blocks with self-attention. The achieved WERs of 1.76/4.37/1.91/4.36 are highly competitive outperforming essentially all models with one exception, the original Conformer. In this case, the Stateformer is able to outper-

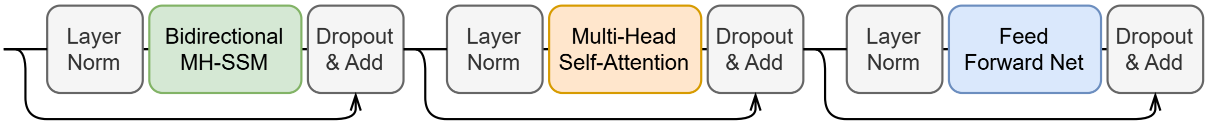

Figure 4: *Stateformer: Bidirectional state space augmented transformer block. It simply has an additional block prior to the attention unit with a bidirectional MH-SSM. The pure MH-SSM architecture is similar but without the self-attention block.*

form the Conformer in all but dev and test-other for which it is competitive. Overall, this demonstrates the power of multi-head state space models both as a standalone model and when combined with self-attention blocks.

5.2. Ablation Studies

5.2.1. Baselines

Table 2 reports the WER performance of smaller baseline models. Overall we can observe that the attention-free S4 (SSM baseline) can outperform the transformer with a time reduction frontend. With a convolutional frontend the larger Conformer is best followed by the transformer. The transformer benefits more from a convolutional frontend as it complements self-attention. Models with convolutional aspects, Conformer and S4 (which can be seen as a variable length convolution) benefit less.

Table 2: *Baseline S4, Transformer and Conformer WER% performance with various frontends (FE). With a CNN frontend, a larger Conformer is best followed by the Transformer.*

Model	Params	FE	dev		test	
			clean	other	clean	other
S4 20L	78.1M	TR	2.45	6.88	2.71	6.72
	78.8M	CNN	2.36	6.60	2.67	6.47
Transformer 20L	76.7M	TR	2.96	7.09	3.05	7.18
	77.5M	CNN	2.45	5.82	2.62	6.15
Conformer 20L	91.9M	TR	2.17	5.54	2.43	5.45
	92.7M	CNN	2.04	5.31	2.26	5.37

5.2.2. Stacking SSM Layers

There are a number of differences between a standard S4 and the MH-SSH block, specifically, the stacking of SSM layers within a single residual block. Table 3 shows the contrast between stacking SSM layers within a residual block versus opting for deeper models. Stacking was found marginally better, and more importantly, allows the model to scale to larger sizes.

Table 3: *Impact of stacking state space layers in a residual block. Stacking 2 layers was found to be marginally more effective than increasing the number of layers.*

Layers	Params	Stack	dev		test	
			clean	other	clean	other
16	56.8M	1	2.57	7.13	2.79	6.86
	66.3M	2	**2.36**	**6.88**	**2.52**	6.59
20	67.6M	1	2.42	6.92	2.67	**6.56**

5.2.3. Number of Heads and IHG

Next, we compare the number of SSM heads and the impact of using IHG instead of standard GLU activations, see Table 4. All of the MH-SSM models use a simpler TR frontend. At 74.7M parameters, the 4H with IHG model is able to outperform the 20L transformer baseline and rival the one with CNN frontend. Overall, the table shows the effectiveness of combining multi-head with head gating.

Table 4: *Ablation study investigating the number of heads and use of IHG in the MH-SSM model. All models use the TR frontend and have 74.7M parameters.*

#Heads	IHG	dev		test	
		clean	other	clean	other
2	✗	2.33	6.92	2.58	6.49
4	✗	2.23	6.74	2.52	6.46
2	✓	2.13	6.81	2.47	6.44
4	✓	2.19	6.38	2.42	6.25
8	✓	2.17	6.49	2.43	6.20

5.2.4. Multi-Scale Frontend

Using the best found MH-SSM from the previous section with 4 heads and IHG enabled, we evaluate the impact of including a multi-scale frontend, see Table 5. With a minor increase of 4.5M parameters, the performance of the MS + MH-SSM system is significantly better. The corresponding Stateformer further improves performance by a notable margin, outperforming the Conformer baseline in Table 2.

Table 5: *Simple comparison of TR vs MS frontend (FE) for the MH-SSM model; including a final Stateformer based on the best found MH-SSM model.*

Model	Params	FE	dev		test	
			clean	other	clean	other
MH-SSM 16L	74.7M	TR	2.19	6.38	2.42	6.25
	79.2M	MS	2.14	6.12	2.39	5.99
Stateformer 16L	96.1M	MS	2.06	5.01	2.27	5.07

6. Conclusions

We proposed a multi-head state space model (MH-SSM) for the audio-encoder of transducer-based speech recognition. Parallel heads of SSMs, together with a novel inter-head gating mechanism was shown highly effective yielding a new class of high-performing models. In addition, we combine MH-SSM blocks and self-attention to form a new type of transformer architecture–the Stateformer. On the LibriSpeech speech recognition task, the proposed MH-SSM outperforms transformer baselines by a large margin. Furthermore, the Stateformer further improves the performance, achieving state-of-the-art performance without external language models.

7. References

[1] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in *International Conference on Acoustics, Speech and Signal Processing*, 2016.

[2] K. Rao, H. Sak, and R. Prabhavalkar, “Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer,” in *Automatic Speech Recognition and Understanding Workshop*, 2017.

[3] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau *et al.*, “Learning phrase representations using RNN encoder–decoder for statistical machine translation,” in *Conference on Empirical Methods in Natural Language Processing*, 2014.

[4] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in *International Conference on Learning Representations*, 2015.

[5] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, “Recurrent neural network based language model,” in *Interspeech*, 2010.

[6] S. Merity, N. S. Keskar, and R. Socher, “Regularizing and optimizing LSTM language models,” in *International Conference on Learning Representations*, 2018.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Conference on Neural Information Processing Systems*, 2017.

[8] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *International Conference on Learning Representations*, 2021.

[10] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang *et al.*, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *International Conference on Computer Vision*, 2021.

[11] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang *et al.*, “Conformer: Convolution-augmented transformer for speech recognition,” *Interspeech*, 2020.

[12] Y. Peng, S. Dalmia, I. Lane, and S. Watanabe, “Branchformer: Parallel mlp-attention architectures to capture local and global context for speech recognition and understanding,” in *International Conference on Machine Learning*, 2022.

[13] K. Kim, F. Wu, Y. Peng, J. Pan, P. Sridhar, K. J. Han, and S. Watanabe, “E-branchformer: Branchformer with enhanced merging for speech recognition,” in *Spoken Language Technology Workshop*, 2022.

[14] A. Gu, I. Johnson, K. Goel, K. Saab, T. Dao, A. Rudra, and C. Ré, “Combining recurrent, convolutional, and continuous-time models with linear state-space layers,” *Conference on Neural Information Processing Systems*, 2021.

[15] A. Voelker, I. Kajić, and C. Eliasmith, “Legendre memory units: Continuous-time representation in recurrent neural networks,” in *Conference on Neural Information Processing Systems*, 2019.

[16] T. Lei, Y. Zhang, S. I. Wang, H. Dai, and Y. Artzi, “Simple recurrent units for highly parallelizable recurrence,” in *Empirical Methods in Natural Language Processing*, 2018.

[17] T. Lei, “When attention meets fast recurrence: Training language models with reduced compute,” *arXiv:2102.12459*, 2021.

[18] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Journal of Basic Engineering*, 1960.

[19] R. J. Hyndman, A. B. Koehler, J. K. Ord, and R. D. Snyder, *Forecasting with Exponential Smoothing: The State Space Approach*. Springer Berlin, Heidelberg, 2008.

[20] J. Durbin and S. J. Koopman, *Time Series Analysis by State Space Methods*. Oxford University Press, 05 2012.

[21] A. Gu, T. Dao, S. Ermon, A. Rudra, and C. Ré, “Hippo: Recurrent memory with optimal polynomial projections,” *Conference on Neural Information Processing Systems*, 2020.

[22] A. Gu, K. Goel, and C. Ré, “Efficiently modeling long sequences with structured state spaces,” in *The International Conference on Learning Representations*, 2022.

[23] H. Mehta, A. Gupta, A. Cutkosky, and B. Neyshabur, “Long range language modeling via gated state spaces,” in *International Conference on Learning Representations*, 2023.

[24] W. L. Brogan, *Modern control theory; 3rd ed.* Englewood Cliffs, NJ: Prentice-Hall, 1991. [Online]. Available: https://cds.cern.ch/record/226422

[25] N. R. Chilkuri and C. Eliasmith, “Parallelizing legendre memory unit training,” in *International Conference on Machine Learning*, 2021.

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

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

[28] K. Goel, A. Gu, C. Donahue, and C. Ré, “It's raw! audio generation with state-space models,” *International Conference on Machine Learning*, 2022.

[29] N. Shazeer, Z. Lan, Y. Cheng, N. Ding, and L. Hou, “Talking-heads attention,” in *arXiv:2003.02436*, 2020.

[30] A. Graves, “Sequence transduction with recurrent neural networks,” in *International Conference of Machine Learning Workshop on Representation Learning*, 2012.

[31] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An asr corpus based on public domain audio books,” in *International Conference on Acoustics, Speech and Signal Processing*, 2015.

[32] D. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. Cubuk, and Q. Le, “Specaugment: A simple data augmentation method for automatic speech recognition,” 2019.

[33] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, “Audio augmentation for speech recognition,” in *Sixteenth annual conference of the international speech communication association*, 2015.

[34] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, “Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss,” 2020.

[35] Y. Wang, A. Mohamed, D. Le, C. Liu, A. Xiao, J. Mahadeokar, H. Huang, A. Tjandra, X. Zhang *et al.*, “Transformer-based acoustic modeling for hybrid speech recognition,” in *International Conference on Acoustics, Speech and Signal Processing*, 2020.

[36] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng *et al.*, “fairseq: A fast, extensible toolkit for sequence modeling,” in *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[37] T. Kudo and J. Richardson, “Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing,” *arXiv:1808.06226*, 2018.

[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[39] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *International Conference for Learning Representations*, 2015.

[40] W. Han, Z. Zhang, Y. Zhang, J. Yu, C.-C. Chiu *et al.*, “Contextnet: Improving convolutional neural networks for automatic speech recognition with global context,” *Interspeech*, 2020.

[41] Y. Zhang, J. Qin, D. S. Park, W. Han, C.-C. Chiu, R. Pang, Q. V. Le, and Y. Wu, “Pushing the limits of semi-supervised learning for automatic speech recognition,” *2010.10504*, 2020.

[42] C. Liu, F. Zhang, D. Le, S. Kim, Y. Saraf, and G. Zweig, “Improving rnn transducer based asr with auxiliary tasks,” in *Spoken Language Technology Workshop*, 2021.