ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10096271

DIAGONAL STATE SPACE AUGMENTED TRANSFORMERS FOR SPEECH RECOGNITION

George Saon, Ankit Gupta and Xiaodong Cui

IBM Research AI, Yorktown Heights, USA

ABSTRACT

We improve on the popular conformer architecture by replacing the depthwise temporal convolutions with diagonal state space (DSS) models. DSS is a recently introduced variant of linear RNNs obtained by discretizing a linear dynamical system with a diagonal state transition matrix. DSS layers project the input sequence onto a space of orthogonal polynomials where the choice of basis functions, metric and support is controlled by the eigenvalues of the transition matrix. We compare neural transducers with either conformer or our proposed DSS-augmented transformer (DSSformer) encoders on three public corpora: Switchboard English conversational telephone speech 300 hours, Switchboard+Fisher 2000 hours, and a spoken archive of holocaust survivor testimonials called MALACH 176 hours. On Switchboard 300/2000 hours, we reach a single model performance of 8.9%/6.7% WER on the combined test set of the Hub5 2000 evaluation, respectively, and on MALACH we improve the WER by 7% relative over the previous best published result. In addition, we present empirical evidence suggesting that DSS layers learn damped Fourier basis functions where the attenuation coefficients are layer specific whereas the frequency coefficients converge to almost identical linearly-spaced values across all layers.

Index Terms— structured state space models, diagonal state space models, neural transducers, end-to-end ASR

1. INTRODUCTION AND RELATED WORK

An interesting alternative to the ubiquitous transformer architecture is the recently introduced structured state space sequence model (S4) which showed promising results for modeling long range dependencies on the LRA (Long Range Arena) benchmark for sequence-level classification of different modalities such as text, images and mathematical expressions [1]. The main idea behind S4 is that the input sequence can be modeled as a linear RNN obtained by discretizing a continuous state space model. The physical meaning of a state in S4 is a time-varying vector of linear expansion coefficients used to approximate the input sequence with orthogonal polynomials under a given measure and support (weighting function and input window) [2]. The appeal of these models is that they can be efficiently implemented as full sequence convolutions running in $\mathcal{O}(T \log T)$ instead of the $\mathcal{O}(T^2)$ complexity for self-attention with T being the input sequence length. Moreover, these models are solidly grounded in function approximation theory and have interpretable parameters in terms of basis functions, measures and time sampling intervals.

In [1] the authors consider a diagonal plus low-rank approximation of the state transition matrix which simplifies the convolutional kernel estimation. In [3], the authors observed that there is no loss in performance when assuming that the transition matrix is diagonal with complex eigenvalues which is conceptually simpler and straightforward to implement compared to [1]. Because of this, diagonal state space (DSS) models will be adopted in this paper. In both works, the authors initialize the diagonal entries of the state transition matrix with the eigenvalues of a higher-order polynomial projection operator (HiPPO) matrix such that the input function is uniformly approximated with Legendre polynomials over a sliding window of fixed length. In [4] the authors argue that parameterizing the eigenvalues in log-space and initializing them with -exp for the real parts and +exp for the imaginary parts is just as effective and improve the DSS model further by augmenting it with self-attention to better capture local dependencies. In [5], the authors revisit the parameterization and initialization of DSS and propose eigenvalue initialization schemes with constant negative real parts with respect to the eigenvalue index and imaginary parts which scale either inversely or linearly with the eigenvalue index. The former results in projecting the input onto the space of Legendre polynomials with uniform weighting from the beginning of the sequence up to the current time whereas the latter amounts to using damped Fourier basis functions as approximators with an exponentially decaying weighted history.

While DSS has been primarily developed as an alternative to self-attention, the dual RNN/convolutional representation suggests that it has potential to outperform the depthwise temporal convolutions in the conformer architecture [6]. We echo the findings of [4] which indicate that self-attention and DSS exhibit complementary behaviour and do not necessarily subsume each other. Given the popularity and effectiveness of conformers for both hybrid [7] and end-to-end ASR [8–12], several other avenues have been explored in the literature to either improve the conformer architecture or the training recipe. In [13], the authors use grouped self-attention and progressive down-sampling to reduce the complexity of the self-attention layer. In [14], the authors provide training recipes and extensive comparisons between conformers and transformers on several corpora. In [15] the authors replace the transformer layer with performer. In [16], the authors use linear self-attention layers. In [7] the authors use two convolutional layers for each conformer block and layer normalization instead of batch norm. Similar to our work, in [17], the authors replace the convolutional layers with a more powerful representation called ConvNeXt.

The main contributions of this work are summarized below:

- We apply diagonal state space models to speech recognition and report experimental results on three public corpora.
- We show that DSSformers outperform conformers when used as encoders for neural transducers and achieve state-of-the-art results for single non-AED models on Switchboard telephony speech and MALACH.
- We study the effect of DSS initialization and provide some insights into what the DSS layers actually learn.

The rest of the paper is organized as follows: in section 2 we review the DSS formalism; in section 3 we present experimental evidence of its utility and in section 4 we summarize our findings.

2. DSS FORMULATION

We briefly review the main concepts behind the diagonal state spaces framework for readers from the ASR community who may not be familiar with this new sequence-to-sequence modeling approach.

2.1. State space model

Borrowing some definitions and notations from [1, 3], a continuous state space model (SSM), sometimes referred to in the literature as a linear time-invariant or a linear dynamical system, is defined by the linear ODE:

$$\begin{aligned} x'(t) &= \mathbf{A}x(t) + \mathbf{B}u(t), \qquad \mathbf{A} \in \mathbb{R}^{N\times N},\ \mathbf{B} \in \mathbb{R}^{N\times 1} \\ y(t) &= \mathbf{C}x(t), \qquad\qquad\qquad\quad \mathbf{C} \in \mathbb{R}^{1\times N} \end{aligned} \tag{1}$$

that maps the continuous 1-dimensional input $u(t) \in \mathbb{R}$ to an N-dimensional latent state $x(t) \in \mathbb{R}^N$ before projecting it to a 1-dimensional output $y(t) \in \mathbb{R}$. The state space is parameterized by the state transition matrix $\mathbf{A}$ as well as trainable parameters $\mathbf{B}, \mathbf{C}$.

2.2. Discretization and link to linear RNNs

Consider a sampling interval $\Delta > 0$ and define $u_k := u(k\Delta), k = 0 \dots L-1$ the sampled input signal. Correspondingly, we have $x_k = x(k\Delta)$ and $y_k = y(k\Delta)$. Equation (1) can be turned into a discrete recurrence that maps $(u_0, \dots, u_{L-1}) \mapsto (y_0, \dots, y_{L-1})$ by integrating over $[(k-1)\Delta, k\Delta]$ under the zero-order hold (ZOH) assumption $u(t) = u_k,\ (k-1)\Delta \le t < k\Delta$:

$$\begin{aligned} x_k &= \overline{\mathbf{A}}x_{k-1} + \overline{\mathbf{B}}u_k, \quad \overline{\mathbf{A}} = e^{\mathbf{A}\Delta},\ \overline{\mathbf{B}} = (e^{\mathbf{A}\Delta} - \mathbf{I})\mathbf{A}^{-1}\mathbf{B} \\ y_k &= \overline{\mathbf{C}}x_k, \qquad\qquad\qquad\qquad\quad \overline{\mathbf{C}} = \mathbf{C} \end{aligned} \tag{2}$$

2.3. Convolutional representation

With the convention $x_{-1} = 0$, the recurrence can be unrolled and rewritten by eliminating the state variables x_k:

$$y_k = \sum_{j=0}^{k} \overline{\mathbf{C}}\overline{\mathbf{A}}^j\overline{\mathbf{B}}u_{k-j}, \qquad k = 0, \dots, L-1 \tag{3}$$

By grouping the scalar coefficients $\overline{\mathbf{C}}\overline{\mathbf{A}}^k\overline{\mathbf{B}}$ into the SSM kernel $\overline{\mathbf{K}} \in \mathbb{R}^L$, $\overline{\mathbf{K}} = (\overline{\mathbf{C}}\overline{\mathbf{B}}, \overline{\mathbf{C}}\overline{\mathbf{A}}\overline{\mathbf{B}}, \dots, \overline{\mathbf{C}}\overline{\mathbf{A}}^{L-1}\overline{\mathbf{B}})$, (3) can be elegantly reformulated as a convolution

$$y = \overline{\mathbf{K}} * u \tag{4}$$

Computing (4) naively would require $\mathcal{O}(L^2)$ operations. Instead, we observe that y_k is the coefficient of z^k of the product $u(z) \cdot \overline{K}(z)$ of two $(L-1)$-degree univariate polynomials $u(z) = \sum_{i=0}^{L-1} u_i z^i$ and $\overline{K}(z) = \sum_{i=0}^{L-1} \overline{K}_i z^i$. By the circular convolution theorem, this product can be computed efficiently in $\mathcal{O}(L \log L)$ using FFT and its inverse.

2.4. Diagonal state spaces

Based on the above, computing y from $\overline{\mathbf{K}}$ and u is easy; the hard part is how to compute $\overline{\mathbf{K}}$ efficiently. The main result in [3] states that if $\mathbf{A} \in \mathbb{C}^{N\times N}$ is diagonalizable over $\mathbb{C}$ with eigenvalues $\lambda_1, \dots, \lambda_N$ such that, $\forall i$, $\lambda_i \neq 0$ and $e^{L\lambda_i\Delta} \neq 1$, there $\exists w \in \mathbb{C}^{1\times N}$ such that

$$\overline{\mathbf{K}} = w \cdot \mathbf{\Lambda}^{-1} \cdot \text{row-softmax}(\mathbf{P}) \tag{5}$$

where $\mathbf{P} \in \mathbb{C}^{N\times L}$, $p_{ik} = \lambda_i k\Delta$ and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_N)$. The proof uses the diagonalization of $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$ which, from the expression of $\overline{\mathbf{A}}$ from (2), implies $\overline{\mathbf{A}}^k = e^{k\mathbf{A}\Delta} = \mathbf{V}e^{k\mathbf{\Lambda}\Delta}\mathbf{V}^{-1}$, and the geometric series identity $\sum_{k=0}^{L-1} z^k = \frac{z^L - 1}{z - 1}$. We refer the reader to [3] for the complete proof.

2.5. DSS layer

A DSS layer operates as follows. It receives an $H \times L$ input sequence and produces an $H \times L$ output sequence where H is the number of channels and L is the sequence length. It does this by applying H DSS kernels to the input (with a shortcut connection) according to (4), one for each coordinate. We apply a Gaussian Error Linear Unit (GELU) nonlinearity to the result followed by an $H \times H$ pointwise linear layer needed to exchange information between the dimensions. After mixing, we apply a Gated Linear Unit (GLU) activation to the output. The implementation of a DSS layer as described so far is publicly available at `https://github.com/ag1988/dss`.

For a state space dimension N, the trainable parameters of the DSS layer are: $\mathbf{\Lambda}_{re}, \mathbf{\Lambda}_{im} \in \mathbb{R}^N$ the diagonal entries of the transition matrix (tied across all channels), $W \in \mathbb{C}^{H\times N}$ from (5), $\Delta \in \mathbb{R}^H$ the time sampling intervals, and $W_{out} \in \mathbb{R}^{H\times H}$ the output mixing matrix.

Just like the depthwise separable convolution module in the conformer architecture, the DSS layer is sandwiched between two pointwise convolutions which serve to increase the inner dimension (typically by a factor of 2) on which the layer operates as shown in Figure 1.

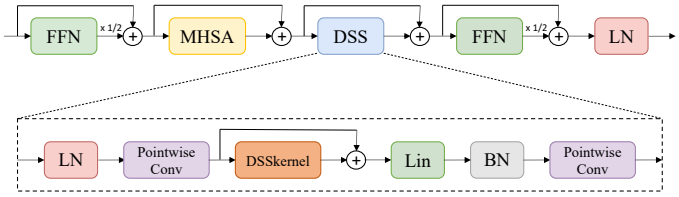


Fig. 1: Proposed architecture: top DSSformer block, bottom DSS module (non-linearities are omitted for clarity).

3. EXPERIMENTS AND RESULTS

We investigate the effectiveness of the proposed model on three public corpora: Switchboard English conversational telephone speech 300 hours, Switchboard+Fisher 2000 hours, and MALACH 176 hours.

3.1. Experiments on Switchboard 300 hours

The acoustic training data comprises 300 hours of English telephone conversations between two strangers on a preassigned topic. We follow the Kaldi `s5c` recipe [18] for data preparation and segmentation and report results on the Hub5 2000 (Switchboard and CallHome),

Hub5 2001 and RT'03 test sets which are processed according to the LDC segmentation and scored using Kaldi WER measurement.

3.1.1. *Feature processing*

Our feature extraction and training recipe largely mirrors [19] with some notable differences. We extract 40-dimensional speaker independent log-Mel features every 10ms with speaker-based mean and variance normalization augmented with Δ and $\Delta\Delta$ coefficients. We perform temporal subsampling by a factor of 2 by stacking every two consecutive frames and skipping every second stacked frame which results in 50 240-dimensional feature vectors per second. Unlike [7, 19, 20], we do not use appended i-vectors as we found them to be less effective with conformer transducers. We create 4 additional replicas of the training data using speed and tempo perturbation [21] both with values in $\{0.9, 1.1\}$ which, together with the original data, amounts to 1500 hours of training data every epoch. We perturb the data in three different ways: (i) sequence noise injection adds, with probability 0.8, a down-scaled spectrum of a random utterance to the current utterance [22]; (ii) SpecAugment randomly masks blocks in both time and frequency with the settings from [23]; (iii) Length perturbation randomly deletes and inserts contiguous frames with probability 0.7 [20].

3.1.2. *Transducer architecture*

We trained neural transducers (or RNN-Ts[1]) [24] with either conformer or DSSformer encoders with 12 layers, feed-forward dimension of 384 and 6×96-dimensional attention heads for an inner dimension of 512. All DSS layers use bidirectional kernels with a state space dimension N=96. The joint network projects the 384-dim vectors from the last encoder layer to 256 and multiplies the result elementwise [19, 25] with a 256-dim projection of a label embedding computed by a unidirectional 1024-cell LSTM prediction network. After the application of hyperbolic tangent, the output is projected to 46 logits followed by a softmax layer corresponding to 45 characters plus BLANK. The baseline conformer RNN-T has an optimal size of 63M parameters and the DSSformer RNN-T has 73M parameters.

3.1.3. *Training and decoding*

The models were trained in Pytorch to minimize the RNN-T loss with CTC loss smoothing from the encoder with a weight of 0.1. Training was carried out on single A100 GPUs for 24 epochs with AdamW SGD and a one cycle learning rate policy which ramps up the step size linearly from 5e-5 to 5e-4 for the first 8 epochs followed by a linear annealing phase to 0 for the remaining 16 epochs. All experiments use a batch size of 64 utterances. Decoding was done using alignment-length synchronous beam search [26]. We also report results with density ratio shallow language model fusion [27] where the target LM is a 12-layer, 512-dimensional transformerXL character LM [28] trained on the Switchboard+Fisher acoustic transcripts (126M characters) and the source LM has the same configuration as the prediction network and was trained on the 300 hours transcripts only (15M characters).

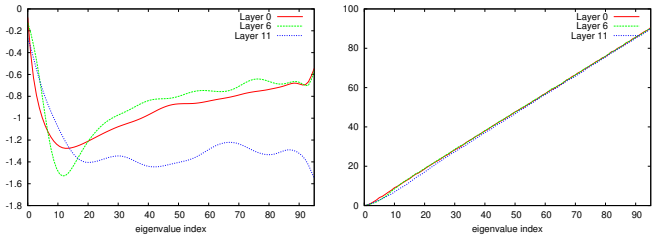
3.1.4. *DSS layer initialization and recognition results*

In Table 1, we compare the performance of baseline conformer and DSSformer transducers with different initializations of the $\mathbf{\Lambda}$ matrix. Concretely, $\mathbf{\Lambda}$ HiPPO uses the top N eigenvalues with positive imaginary part from the skew-symmetric $2N \times 2N$ matrix $a_{ij} = \begin{cases} 2(i+1)^{1/2}(2j+1)^{1/2}, & i<j \\ -1/2, & i=j \\ -2(i+1)^{1/2}(2j+1)^{1/2}, & i>j \end{cases}$ [3]. For $\mathbf{\Lambda}$ exp random, $\lambda_n = -e^{a_n} + i \cdot e^{b_n}$ where $a_n, b_n \sim \mathcal{U}[-1, 1]$ [4]. For $\mathbf{\Lambda}$ S4D-Inv, $\lambda_n = -\frac{1}{2} + i\frac{N}{\pi}\left(\frac{N}{2n+1} - 1\right)$, whereas for $\mathbf{\Lambda}$ S4D-Lin, $\lambda_n = -\frac{1}{2} + i\pi n$ [5]. For all experiments, Δ is parameterized in log-space with values drawn from $\mathcal{U}[\log(0.001), \log(0.1)]$ and the real and imaginary parts for w in (5) are initialized from $\mathcal{N}(0, 1)$.

Encoder	Hub5'00			Hub5'01	RT'03
	swb	ch	avg		
Conformer	7.5	15.0	11.2	11.2	14.4
-MHSA+DSS	8.0	15.9	12.0	12.2	15.7
$\mathbf{\Lambda}$ HiPPO [3]	7.2	14.2	10.7	10.7	13.5
$\mathbf{\Lambda}$ exp random [4]	7.3	14.5	10.9	10.8	13.4
$\mathbf{\Lambda}$ S4D-Inv [5]	7.5	14.7	11.1	10.9	13.8
$\mathbf{\Lambda}$ S4D-Lin [5]	7.2	14.3	10.8	10.9	13.3
$\lambda_n = -1 + i \cdot n$	7.1	13.9	10.5	10.5	13.3

Table 1: Recognition results for conformer, conformer with MHSA replaced by DSS, and DSSformer transducers with different $\mathbf{\Lambda}$ initializations on Switchboard 300 hours (Hub5'00, Hub5'01, RT'03 test sets). All models are trained for 24 epochs without length perturbation and decodings are done without external LM.

The $\mathbf{\Lambda}$ initialization from the last row in Table 1 was motivated by inspecting the converged values of λ_n when the DSS layers were initialized with S4D-Lin. Interestingly, the imaginary parts of λ_n converge from πn to approximately $0.95n$ across all layers as shown in Figure 2b. In contrast, in Figure 2a the real parts converge to values that are layer-dependent[2]. This suggests that the DSS layers learn damped Fourier basis functions $F_n(t) = e^{-\lambda_n t}$ where the attenuation coefficients are layer specific and the frequency coefficients are linearly spaced and common across layers. The benefit of using FFT layers for mixing input sequences has also been shown in the FNet architecture [29].



(a) Real parts of trained λ_n (b) Imaginary parts of trained λ_n

Fig. 2: Converged eigenvalues for S4D-Lin initialization for first, middle and last layers on Switchboard 300 hours.

In Table 2 we compare the performance of our best single DSSformer model with existing approaches from the literature. Here, the model was trained for 30 epochs with length perturbation with

[1]Both terms are used interchangeably in the literature even for models where the encoder is not an RNN.

[2]The curves have been smoothed with Bezier interpolation for ease of visualization.

the following settings from [20]: insertion and deletion probabilities of 0.7, 10% of frames selected as starting points for both, maximum deletion length of 7 frames and maximum insertion length of 3 frames. Length perturbation is lifted after 25 epochs.

Work	Model	Encoder	LM	Hub5'00			Hub5'01
				swb	ch	avg	
[14]	AED	Conformer	–	7.1	15.0	11.1	–
[8]	AED	Conformer	–	6.7	13.0	9.9	10.0
			LSTM*	5.7	11.4	8.6	8.5
			+Trafo*	5.5	11.2	8.4	8.5
[7]	HMM	Conformer	n-gram	7.1	13.5	10.3	10.4
			Trafo	6.3	12.1	9.2	9.3
[20]	RNN-T	LSTM	–	6.9	14.5	10.7	11.2
			LSTM	5.9	12.5	9.2	9.4
[30]	RNN-T	Conformer	n-gram	–	–	10.3	10.6
			Trafo	–	–	9.3	9.4
Ours	RNN-T	DSSformer	–	6.7	13.4	10.0	10.3
			Trafo	5.6	12.2	8.9	9.0

Table 2: Performance comparison of DSSformer transducer with other single-model approaches from the literature on Switchboard 300 hours (* are cross-utterance LMs).

3.2. Experiments on Switchboard+Fisher 2000 hours

The second set of experiments was carried out on 1975 hours (9875 hours after augmentation) comprised of 262 hours of Switchboard 1 audio with segmentations and transcripts provided by Mississippi State University plus 1698 hours from the Fisher data collection with transcripts provided by LDC plus 15 hours of CallHome audio. We trained neural transducers with either conformer (10 or 12 layers) or DSSformer encoders (10 layers), feed-forward dimension of 512 and 8×64-dimensional attention heads. All DSS layers use bidirectional kernels with a state space dimension N=96. Training was carried out on 4 A100 GPUs with an effective batch size of 128 for 20 epochs with a one cycle LR policy with a maximum learning rate of 5e-4. The other settings are the same as in 3.1. In Table 3 we show a comparison of baseline conformer and DSSformer transducers with various $\mathbf{\Lambda}$ initializations. As can be seen, DSSformer encoders outperform the conformer counterparts and the best $\mathbf{\Lambda}$ initialization is the same as in 3.1. For contrast, we also compare our results with the single best performing model on this task from [8] and note that we achieve a comparable performance on two out of three test sets.

Encoder	Hub5'00			Hub5'01	RT'03
	swb	ch	avg		
Conformer (10L)	5.2	8.5	6.9	7.6	7.8
Conformer (12L)	5.4	8.5	6.9	7.6	8.2
$\mathbf{\Lambda}$ HiPPO	5.2	8.4	6.8	7.4	7.5
$\mathbf{\Lambda}$ S4D-Lin	5.3	8.4	6.8	7.6	7.5
$\lambda_n = -1 + i \cdot n$	5.1	8.5	6.8	7.4	7.4
+length perturb.	5.2	8.2	6.7	7.2	7.5
Conformer AED [8]	4.8	8.0	6.4	7.3	7.5

Table 3: Recognition results for conformer (10 and 12 layers) and DSSformer transducers (10 layers) with different $\mathbf{\Lambda}$ initializations on Switchboard 2000 hours (Hub5'00, Hub5'01, RT'03 test sets). All decodings are done without external LM.

3.3. Experiments on MALACH 176 hours

Lastly, we test the proposed models on the public MALACH corpus [31] (released by LDC as LDC2019S11) which consists of Holocaust testimonies collected by the Survivors of the Shoah Visual History Foundation. The corpus is 16kHz audio broken down into 674 conversations totaling 176 hours for training (880 hours after augmentation) and 8 conversations of 3.1 hours for testing. The collection consists of unconstrained, natural speech filled with disfluencies, heavy accents, age-related coarticulations, un-cued speaker and language switching, and emotional speech, all of which present significant challenges for current ASR systems. Because of this, the error rates reported are significantly higher than for the previous corpora. We trained conformer and DSSformer transducers with the same feature extraction, architecture, DSS layer initialization and training recipe as in 3.1 without length perturbation and with S4D-Lin $\mathbf{\Lambda}$ initialization. In Table 4 we report results with and without external LM fusion where the LM is a 10 layer 512-dimensional transformerXL trained on 7.2M characters. Our results show a 7% relative improvement in WER over the previous best hybrid LSTM approach.

Work	Model	Encoder	LM	WER
[31]	HMM	GMM	n-gram	32.1
[32]	HMM	LSTM	n-gram	23.9
			LSTM	21.7
Ours	RNN-T	Conformer	–	21.5
		DSSformer	–	20.9
			Trafo	20.2

Table 4: Performance comparison of conformer and DSSformer transducer with other single-model approaches from the literature on MALACH 176 hours.

4. DISCUSSION

Diagonal state space models are a promising alternative to temporal convolutions with fixed-length kernels for ASR when used in a conformer-style architecture. We attribute their success to the connection with function approximation theory and to the interpretability of their parameters. In future work we will investigate better ways of integrating DSS layers with self-attention and feedforward modules as opposed to simply using them as a drop-in replacement for the depthwise convolutions in conformer. For example, the DSS mixing matrix can be combined with the second pointwise convolution which will simplify the overall architecture. Another avenue of research is to improve the initialization for the real parts of the eigenvalues of the state transition matrices and possibly keep the $\mathbf{\Lambda}$s fixed during training which will reduce the number of free parameters. Lastly, we plan to study the effectiveness of DSS for other end-to-end ASR modeling approaches.

References

[1] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," *ArXiv preprint*, vol. abs/2111.00396, 2021.

[2] A. Gu, T. Dao, S. Ermon, et al., "Hippo: Recurrent memory with optimal polynomial projections," in *Advances in Neural Information Processing Systems 33: Annual Conference on*

Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, H. Larochelle, M. Ranzato, R. Hadsell, et al., Eds., 2020.

[3] A. Gupta, "Diagonal state spaces are as effective as structured state spaces," *ArXiv preprint*, vol. abs/2203.14343, 2022.

[4] H. Mehta, A. Gupta, A. Cutkosky, and B. Neyshabur, "Long range language modeling via gated state spaces," *ArXiv preprint*, vol. abs/2206.13947, 2022.

[5] A. Gu, A. Gupta, K. Goel, and C. Ré, "On the parameterization and initialization of diagonal state space models," *ArXiv preprint*, vol. abs/2206.11893, 2022.

[6] A. Gulati, J. Qin, C. Chiu, et al., "Conformer: Convolution-augmented transformer for speech recognition," in *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, H. Meng, B. Xu, and T. F. Zheng, Eds. 2020, pp. 5036–5040, ISCA.

[7] M. Zeineldeen, J. Xu, C. Lüscher, et al., "Improving the training recipe for a robust conformer-based hybrid model," *ArXiv preprint*, vol. abs/2206.12955, 2022.

[8] Z. Tüske, G. Saon, and B. Kingsbury, "On the limit of english conversational speech recognition," in *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, H. Hermansky, H. Cernocký, L. Burget, et al., Eds. 2021, pp. 2062–2066, ISCA.

[9] T. N. Sainath, Y. He, A. Narayanan, et al., "Improving the latency and quality of cascaded encoders," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8112–8116.

[10] J. Li et al., "Recent advances in end-to-end automatic speech recognition," *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 1, 2022.

[11] Y. Shi, C. Wu, D. Wang, et al., "Streaming transformer transducer based speech recognition using non-causal convolution," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8277–8281.

[12] Y. Zhang, D. S. Park, W. Han, et al., "Bigssl: Exploring the frontier of large-scale semi-supervised learning for automatic speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, 2022.

[13] M. Burchi and V. Vielzeuf, "Efficient conformer: Progressive downsampling and grouped attention for automatic speech recognition," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 8–15.

[14] P. Guo, F. Boyer, X. Chang, et al., "Recent developments on espnet toolkit boosted by conformer," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5874–5878.

[15] P. Wang and D. Wang, "Efficient end-to-end speech recognition using performers in conformers," *arXiv e-prints*, pp. arXiv–2011, 2020.

[16] S. Li, M. Xu, and X.-L. Zhang, "Efficient conformer-based speech recognition with linear attention," in *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2021, pp. 448–453.

[17] Y. Jiang, J. Yu, W. Yang, et al., "Nextformer: A convnext augmented conformer for end-to-end speech recognition," *ArXiv preprint*, vol. abs/2206.14747, 2022.

[18] D. Povey, A. Ghoshal, G. Boulianne, et al., "The Kaldi speech recognition toolkit," in *IEEE workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011.

[19] G. Saon, Z. Tüske, D. Bolanos, and B. Kingsbury, "Advancing rnn transducer technology for speech recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5654–5658.

[20] X. Cui, G. Saon, T. Nagano, et al., "Improving generalization of deep neural network acoustic models with length perturbation and n-best based label smoothing," *ArXiv preprint*, vol. abs/2203.15176, 2022.

[21] 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.

[22] G. Saon, Z. Tüske, K. Audhkhasi, and B. Kingsbury, "Sequence noise injected training for end-to-end speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*. 2019, pp. 6261–6265, IEEE.

[23] D. S. Park, W. Chan, Y. Zhang, et al., "Specaugment: A simple data augmentation method for automatic speech recognition," in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, G. Kubin and Z. Kacic, Eds. 2019, pp. 2613–2617, ISCA.

[24] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[25] C. Zhang, B. Li, Z. Lu, et al., "Improving the fusion of acoustic and text representations in rnn-t," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8117–8121.

[26] G. Saon, Z. Tüske, and K. Audhkhasi, "Alignment-length synchronous decoding for RNN transducer," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*. 2020, pp. 7804–7808, IEEE.

[27] E. McDermott, H. Sak, and E. Variani, "A density ratio approach to language model fusion in end-to-end automatic speech recognition," in *Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 434–441.

[28] Z. Dai, Z. Yang, Y. Yang, et al., "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019, pp. 2978–2988, Association for Computational Linguistics.

[29] J. Lee-Thorp, J. Ainslie, I. Eckstein, and S. Ontanon, "Fnet: Mixing tokens with fourier transforms," *arXiv preprint arXiv:2105.03824*, 2021.

[30] W. Zhou, W. Michel, R. Schlüter, and H. Ney, "Efficient training of neural transducer for speech recognition," *ArXiv preprint*, vol. abs/2204.10586, 2022.

[31] B. Ramabhadran, J. Huang, and M. Picheny, "Towards automatic transcription of large spoken archives - English ASR for the MALACH project," in *ICASSP*, 2003.

[32] M. Picheny, Z. Tüske, B. Kingsbury, et al., "Challenging the boundaries of speech recognition: The MALACH corpus," in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, G. Kubin and Z. Kacic, Eds. 2019, pp. 326–330, ISCA.