Interspeech 2024
1-5 September 2024, Kos, Greece

Exploring the Capability of Mamba in Speech Applications

Koichi Miyazaki[1], *Yoshiki Masuyama*[1,2], *Masato Murata*[1]

[1]CyberAgent, Inc., Japan
[2]Tokyo Metropolitan University, Japan
miyazaki_koichi_xa@cyberagent.co.jp

Abstract

This paper explores the capability of Mamba, a recently proposed architecture based on state space models (SSMs), as a competitive alternative to Transformer-based models. In the speech domain, well-designed Transformer-based models, such as the Conformer and E-Branchformer, have become the de facto standards. Extensive evaluations have demonstrated the effectiveness of these Transformer-based models across a wide range of speech tasks. In contrast, the evaluation of SSMs has been limited to a few tasks, such as automatic speech recognition (ASR) and speech synthesis. In this paper, we compared Mamba with state-of-the-art Transformer variants for various speech applications, including ASR, text-to-speech, spoken language understanding, and speech summarization. Experimental evaluations revealed that Mamba achieves comparable or better performance than Transformer-based models, and demonstrated its efficiency in long-form speech processing.
Index Terms: Automatic speech recognition, text-to-speech, state-space model, long-range dependency

1. Introduction

Speech processing has witnessed significant performance improvements with the recent progress in end-to-end sequence-to-sequence models [1–3]. The key to this advancement is Transformer [4] with self-attention that captures the global context and allows parallel training. Consequently, the performance of various speech processing tasks has been improved over long short-term memory-based models [5]. Building upon the success of Transformers, several variants have been tailored for speech processing tasks [6–8]. Conformer [6] and E-Branchformer [7] have become de facto standards in this field. Conformer combines a self-attention mechanism with convolutional neural networks to capture both global and local contextual information in a cascaded manner. This hybrid architecture demonstrated superior performance in automatic speech recognition (ASR) and text-to-speech (TTS). Meanwhile, E-Branchformer adopts a parallel structure to extract global and local contexts, and subsequently merges the outputs of the two branches. E-Branchformer has achieved state-of-the-art results on various benchmarks [9].

Despite the success of Transformers, they suffer from the quadratic time and memory complexity in the vanilla attention mechanism. This prevents models from scalability to long-form speech and motivates exploring alternative to Transformers [10, 11]. Towards efficient modeling of long-range dependencies, neural state space models (SSMs) have emerged as a promising alternative architecture. In particular, SSMs leverage a state to represent the past sequences instead of attending the entire sequences at each time step. This procedure can be performed in parallel with a sub-quadratic complexity by using tailored algorithms. SSMs have exhibited promising results in several speech processing tasks, such as ASR [12–15], speech synthesis [16], and speech enhancement [17, 18]. Most SSMs, including S4 [19], have mimicked a time-invariant system, which inherently limiting their ability for input-dependent processing represented by selective copying and induction heads [20].

To overcome this limitation, Mamba introduced a selection mechanism that parameterizes the SSM parameters based on the input-dependent processing [20]. This modification has yielded promising results in various fields such as natural language processing (NLP) [21], and computer vision (CV) [22]. Although Mamba seems to be competitive alternative to Transformers, its superiority in general speech applications was not comprehensively validated in the original paper [20].

In this paper, we investigate the efficacy of Mamba on various end-to-end speech processing applications: ASR, TTS, spoken language understanding (SLU), and speech summarization (SUMM). Our experimental evaluation covers various types of tasks and long-form speech processing. Experimental results show that Mamba achieves comparable performance to Conformer/E-Branchformer on various datasets, even outperforming Transformers on long-form speech processing. We also stress that Mamba is directly applicable to long-form SUMM due to its sub-quadratic complexity, even when Conformer faces the issue of out-of-memory.

2. Mamba

In this section, we briefly review Mamba, which is a recently proposed SSM, as a promising sequence-to-sequence model. SSMs are inspired by well-established frameworks such as Kalman filters and hidden Markov models [19, 23, 24]. An SSM maps an input sequence $\mathbf{x} \in \mathbb{R}^D$ to $\mathbf{y} \in \mathbb{R}^D$ in an element-wise manner. In detail, the discrete SSM with element-wise latent state $\mathbf{h}_d \in \mathbb{R}^N$ can be formulated as follows [1]:

$$\mathbf{h}_{t,d} = \bar{\mathbf{A}}\mathbf{h}_{t-1,d} + \bar{\mathbf{B}}x_{t,d}, \tag{1}$$

$$y_{t,d} = \mathbf{C}\mathbf{h}_{t,d}, \tag{2}$$

$$\bar{\mathbf{A}}, \bar{\mathbf{B}} = \exp(\Delta\mathbf{A}), \Delta\mathbf{B}, \tag{3}$$

where $\mathbf{A} \in \mathbb{R}^{N\times N}$, $\mathbf{B} \in \mathbb{R}^{N\times 1}$, $\mathbf{C} \in \mathbb{R}^{1\times N}$, and $\Delta \in \mathbb{R}_+$ represent continuous SSM parameters [19]. We assume that $\mathbf{A}$ is diagonal for simplicity and computational efficiency. One of the key contributions of Mamba is introducing a selection mechanism that allows SSM to be input-dependent. That is, rather than directly optimizing the SSM parameters, we optimize additional parameters ($\mathbf{W}_B$, $\mathbf{W}_C$, and $\mathbf{W}_\Delta$) that compute the SSM

[1]Note that [20] explains zero-order hold discretization; however, we used an approximated version followed by the official implementation.

10.21437/Interspeech.2024-994

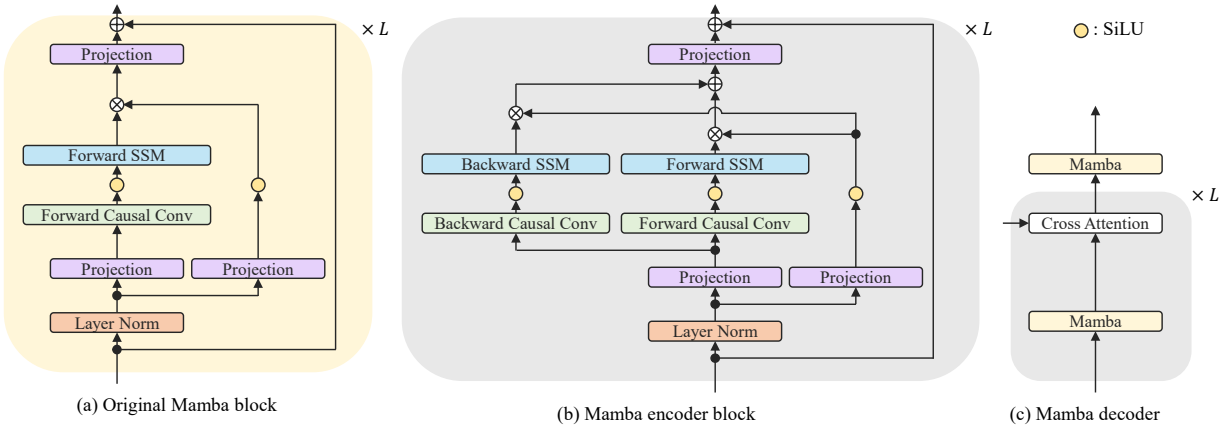

Figure 1: *Architecture of the Mamba block. (a) The original Mamba block. (b) The Mamba encoder block extends the original Mamba block in a bidirectional design. This modification allows for capturing past and future contexts in the input sequences. (c) The Mamba decoder. To bridge the encoder output, we employed a cross-attention after an original Mamba block.*

parameters as follows:

$$\mathbf{B}_t, \mathbf{C}_t = \mathbf{W}_B \mathbf{x}_t, (\mathbf{W}_C \mathbf{x}_t)^{\mathsf{T}}, \quad (4)$$

$$\Delta_t = \text{softplus}(\mathbf{W}_\Delta \mathbf{x}_t), \quad (5)$$

where $\mathbf{W}_B \in \mathbb{R}^{N \times D}$, $\mathbf{W}_C \in \mathbb{R}^{N \times D}$, $\mathbf{W}_\Delta \in \mathbb{R}^{1 \times D}$, softplus function refers to $\log(1 + \exp(x))$, and $(\cdot)^{\mathsf{T}}$ denotes the transpose. By making the SSM input-dependent, however, the efficient algorithms that utilize global convolution [19] are no longer applicable. To address this limitation, Mamba uses parallel scan [24, 25] and hardware-efficient algorithms.

In this study, we used Mamba in the encoder and decoder, where we modified the original Mamba block as illustrated in Figure 1. Each block had a layer normalization [26] and residual connection. In the inner block, the input signal mapped $\mathbb{R}^{D_{\text{in}}} \mapsto \mathbb{R}^{ED_{\text{in}}}$ through an input projection layer, then applied causal convolution and SSM layer. Thereafter, the processed signal mapped back $\mathbb{R}^{ED_{\text{in}}} \mapsto \mathbb{R}^{D_{\text{in}}}$ through an output projection layer. As the SSM has a causal operation, we extended bidirectional design similar, to that ofbidirectional RNNs [27]. To bridge the encoder output to the decoder, we employ cross-attention after each Mamba block. Note that SSM handles positional information implicitly, we removed the positional encoding used in the Transformer-based encoder and decoder. We employed the official Mamba codebase[2]. We used the S4D-Real [23] initialization such that $\mathbf{A}_n = -(n+1)$, state size N set to 16, expansion factor E set to 4, and initial Δ parameters are uniformly sampled from $[0.001, 0.1]$ for all experiments.

3. Automatic Speech Recognition

3.1. Setups

Data. We used seven diverse ASR datasets that covered various languages, speaking styles, and a range of dataset sizes. The evaluation metrics and dataset split follow the ESPnet recipes[3].
Models. We compared the attention encoder-decoder (AED) model with different combinations of encoder (E-Branchformer or Mamba) - decoder (Transformer, S4, or Mamba) architectures. We explored these combinations to gain insights into the effectiveness of Mamba as an encoder and/or decoder for ASR tasks. For a fair comparison, we set the expansion factor to four in both the Mamba encoder and decoder, resulting in a similar model sizes. For the small models, the E-Branchformer encoder had 12 blocks, whereas the Mamba encoder had 24 blocks. For the large models, the E-Branchformer encoder had 17 blocks, whereas the Mamba encoder has 30 blocks. Transformer and Mamba decoder had six blocks in all experiments.

[2] https://github.com/state-spaces/mamba
[3] https://github.com/espnet/espnet

Table 1: *WER (%) for different encoder and decoder architectures on LibriSpeech 100h test sets. CTC and AED are performed with greedy search and beam search, respectively. Real-time factor (RTF) is calculated using a single A100 GPU. All results are obtained without an external language model.*

Model			Results↓		
Encoder	Decoder	Params	clean	other	RTF
CTC					
Transformer	N/A	17.3	12.8	28.1	0.118
Conformer	N/A	27.0	9.8	23.3	0.193
E-Branchformer	N/A	26.4	9.5	**22.9**	0.189
(uni-)Mamba	N/A	24.2	15.7	33.6	**0.117**
(bi-)Mamba	N/A	26.3	**9.1**	23.5	0.152
AED					
E-Branchformer	Transformer	38.5	6.4	17.0	0.453
E-Branchformer	S4	34.9	6.3	16.5	0.360
E-Branchformer	(uni-)Mamba	36.7	**6.1**	**16.5**	0.357
(bi-)Mamba	Transformer	38.3	6.6	18.9	0.351
(bi-)Mamba	S4	34.8	6.5	18.6	0.349
(bi-)Mamba	(uni-)Mamba	36.6	6.5	18.5	**0.346**

Training. We followed the ESPnet recipes for data preparation, training, decoding, and evaluation. The data augmentation and hyper-parameters were the same as those provided in the recipe for the E-Branchformer. We observed that the Mamba model was sometimes unstable during training and tended to overfit. We thus added dropout regularization after the input and output projection layers. To stabilize training, we used AdamW optimizer [34] instead of Adam [35] and set a stricter dropout probability of 0.2. All models were trained on A100 GPUs.

3.2. Results

Table 1 lists the ASR results for various combinations of encoders and decoders on the LibriSpeech 100h dataset. We compared the encoder-only models using CTC. The CTC re-

Table 2: *ASR results on various datasets using hybrid CTC/Attention model. Token refers to the input and output token type. Char and BPE represent character and byte pair encoding, respectively. Params refers to the total number of parameters ($\times 10^6$). † means the result with a shallow fusion of a Transformer language model. ‡ means Conformer encoder is used instead of E-Branchformer as the training was failed. Note that E-Branchformer-Transformer is reproduced by the provided recipe in ESPnet2.*

Dataset	Token	Metric	Evaluation Sets	E-Branchformer-Transformer		E-Branchformer-Mamba	
				Params	Results ↓	Params	Results ↓
AISHELL [28]	Char	CER	dev / test	45.7	**4.2 / 4.4**	43.9	**4.2** / 4.6
GigaSpeech [29]	BPE	WER	dev / test	148.9	**10.5 / 10.6**	153.7	10.7 / 10.7
CSJ [30]	Char	CER	eval1 / eval2 / eval3	146.3	**3.5 / 2.7 / 2.9**	151.1	3.7 / 2.8 / **2.9**
LibriSpeech 100h [31]	BPE	WER	{dev,test}_{clean,other}	38.5	6.3 / 17.0 / 6.4 / 17.0	36.7	**6.0 / 16.2 / 6.1 / 16.5**
LibriSpeech 960h [31]	BPE	WER	{dev,test}_{clean,other}	148.9	† **1.7 / 3.6** / 1.9 / **3.9**	153.7	† 1.7 / **3.6 / 1.8 / 3.9**
TEDLIUM2 [32]	BPE	WER	dev / test	35.0	‡ 9.1 / 7.4	33.3	‡ **8.1 / 7.3**
VoxForge [33]	Char	CER	dt_it / et_it	34.7	8.6 / 8.1	32.9	**8.5 / 8.0**

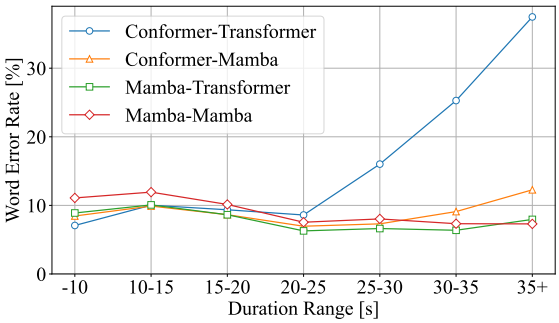

Figure 2: *Long-form ASR results on TEDLIUM2.*

sults reveal that making Mamba bi-directional leads significant performance improvements. However, it did not surpass the Conformer/E-Branchformer in test_other. This suggests the Mamba model may be less robust in encoding acoustic features. The AED results demonstrate that using an SSM-based decoder yields a better performance compared to the Transformer decoder. Table 2 lists ASR results across various benchmarks. We compared the E-Branchformer-Transformer with the E-Branchformer-Mamba, which performs the best, as summerized in Table 1. E-Branchformer-Mamba performed comparably to E-Branchformer-Transformer in various scenarios.

3.3. Discussion

Figure 2 shows the performance of long-form speech recognition. The models were trained using the TEDLIUM2 dataset and evaluated using three concatenated consecutive speech segments. The Conformer-Transformer exhibited a gradual degradation in recognition performance as the length of the evaluated speech exceeded 25 s. We successfully mitigate this performance degradation by incorporating Mamba into either the encoder or decoder, This observation is consistent with recent findings that highlighting the challenges associated with positional embedding in processing long sequence lengths. Therefore, it supports the efficacy of the models employing SSMs for enhanced long-form speech recognition.

4. Text-to-Speech

To investigate the capability of Mamba for generative tasks, we performed TTS experiments using LJSpeech [36] dataset.

4.1. Setups

Data. LJSpeech dataset contains 24 hours of audiobook speech uttered by a single female speaker. The dataset had 13,100 utterances, and we split the dataset into 12600/250/250 utterances each for training, development, and evaluation set, respectively.
Models. We used Conformer-FastSpeech2 [37] as the baseline, which is an extension of the FastSpeech2 [38] model using Conformer blocks instead of Transformer blocks. To assess the effectiveness of Mamba in TTS, we integrated Mamba blocks into the FastSpeech2 backbone (Mamba-FastSpeech2). Since FastSpeech2 is a non-autoregressive model, we used a bidirectional Mamba for both the encoder and decoder modules. To synthesize waveform from the generated acoustic features, we use a HiFi-GAN vocoder [39]. The HiFi-GAN model is trained using the same dataset split as the baseline.
Training. We followed the ESPnet2 recipe but reduced the total training steps from 1.0×10^6 to 1.0×10^5. This reduction is sufficient to achieve reasonable quality while significantly reducing the computational cost and training time. We used g2p_en[4] as a grapheme-to-phoneme function. Conformer-FastSpeech2 has four Conformer blocks in the encoder and decoder modules. Mamba-FastSpeech2 has eight Mamba encoder blocks in the encoder and decoder modules. We used the duration labels extracted using the Montreal Forced Aligner toolkit [40].

4.2. Results

We conducted subjective and objective evaluations to assess speech quality. For the subjective evaluation, we performed a 5-scale mean opinion score (MOS) test and a preference test via Amazon Mechanical Turk with 50 participants. They evaluated randomly selected 50 audio samples from each method and five random audio pairs with the same utterance. For the objective evaluation, we followed the evaluation process in ESPnet-TTS, including the mel-cepstral distortion (MCD), log-F_o root-mean-square error (logF_o), and CER. Table 3 and Figure 3 show the TTS results, indicating that Mamba-FastSpeech2 has comparable performance on both subjective and objective evaluations.

5. Spoken language understanding

To demonstrate the effectiveness of Mamba in predicting high-level semantics, we performed SLU experiments on SLURP [41] and SLUE [42] following ESPnet-SLU [43].

5.1. Setups

Data. SLURP [41] contains utterances of single-turn user interactions with a home assistant, where each recording is annotated with a scenario, action, and entities. We performed in-

[4] https://github.com/Kyubyong/g2p

Table 3: *TTS results. GT (mel) refers to a reconstructed sample using ground truth mel-spectrogram with HiFi-GAN vocoder. CFS2 and MFS2 denote the synthesized speech samples from Conformer-FastSpeech2 and Mamba-FastSpeech2, respectively. CI represents the 95 % confidence interval*

Method	MCD [dB] ↓	logF_o ↓	CER (%)↓	MOS ↑ ±CI
GT(mel)	3.75	0.156	1.1	4.06 ± 0.11
CFS2	**6.51**	**0.217**	**1.7**	3.72 ± 0.12
MFS2	6.54	0.219	1.9	$\mathbf{3.76 \pm 0.12}$

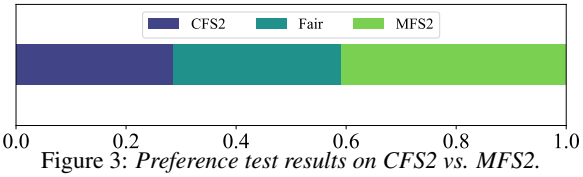

Figure 3: *Preference test results on CFS2 vs. MFS2.*

tent classification and entity prediction, where the intent corresponds to the scenario and action pair. SLUE [42] is a low-resource benchmark for sentiment analysis and named entity recognition. As the annotations for the evaluation set are not public, we report the results for the development set.

Models. AED models were used to jointly predict the label for SLU and the corresponding transcription [43]. We investigated the performance using different encoder architectures: Conformer, E-branchformer, and Mamba. We used the popular log-mel filter-bank as the front-end. The baseline model followed the configurations adopted in [9] and [44] for SLURP and SLUE, respectively. The Mamba encoder followed a small model in the ASR experiment and aligned the number of parameters to those of the baseline models. The Transformer decoder consisted of six blocks.

Training. We followed the ESPnet recipes for data preparation, training, and inference. For SLURP, we set the maximum learning rate to 2.0×10^{-4} with Adam optimizer following the baseline configuration. For SLUE, we used AdamW optimizer and adjusted the learning rate to 2.0×10^{-3} and 5.0×10^{-4} for sentiment analysis and named entity recognition, respectively.

5.2. Results

Table 4 compares the SLU results for the different encoder architectures. Mamba consistently performed better than Transformers with a similar model sizes. In particular, on SLURP, we computed the confidence intervals for intent accuracy using the official Python toolkit. We confirmed that the confidence interval for Mamba was $(87.5, 88.7)$, while that for E-Branchfromer was $(86.2, 87.4)$. This result indicates that Mamba has the potential to outperform Transformers in semantic tasks.

6. Speech summarization

End-to-end abstractive speech summarization aims to generate a short summary from a long-form speech, which requires modeling the long-range dependencies [45,46]. Since we need to handle long-form speech in the encoder, the quadratic complexity of the attention mechanism is problematic. Hence, we explore the benefit of Mamba in the encoder.

6.1. Setups

Data. We used the How2 corpus [47] containing 2000 hours YouTube videos and their descriptions. Utterance-level and entire video-level sub-sets were provided for ASR and SUMM, respectively. As described in [48], we used 40-dimensional filterbank and 3-dimensional pitch features were used as inputs.

Table 4: *SLU results with different encoders. SLURP shows intent accuracy (%) and SLU-F1 (%). SLUE shows macro F1 (%) for sentiment analysis and named entity recognition.*

Dataset	Conformer	E-branchformer	Mamba
SLURP	86.1 / 77.4	86.8 / 78.0	**88.1 / 78.3**
SLUE	34.2 / 39.4	33.5 / 40.5	**35.0 / 41.9**

Table 5: *SUMM results on How2 dataset.*

Method	R-L ↑	MTR ↑	BSc ↑
Conformer-Transformer (100s)	60.5	32.2	92.5
Mamba-Transformer (100s)	62.3	33.5	92.9
Mamba-Transformer (600s)	62.9	33.8	93.1
Mamba-Mamba (600s)	62.7	**33.9**	93.0
Conformer-Transformer (100s) [48]	62.3	30.4	93.0
FNet-Transformer [48]	**64.0**	32.7	**93.5**

Models. We replaced the Conformer encoder in the baseline AED with a 24-block Mamba encoder, as its attention mechanism is the main bottleneck in computational complexity. The number of parameters was aligned with the baseline models. The Transformer decoder comprised six blocks for both models. Our entire model had 96.4×10^6 parameters, whereas the baseline Conformer model had 97.7×10^6 parameters. We also investigated the performance by replacing the Transformer decoder with a Mamba decoder.

Training. We pre-trained AED on utterance-level ASR before fine-tuning on SUMM over entire video [48]. We pre-trained the Mamba encoder with the AdamW optimizer, where the peak learning rate was 2.0×10^{-3}. The AED was subsequently fine-tuned on SUMM with an initial learning rate of 1.0×10^{-4}. During the fine-tuning of the Conformer model, we trimmed input audio at 100 s owing to the out-of-memory issue on A100 [45, 48]. In contrast, our Mamba model can leverage 600 s audio thanks to its sub-quadratic complexity.

6.2. Results

We evaluated the generated summarization by ROUGE-L (R-L) [49], METOR (MTR) [50], and BERTScote (BSc) [51], as proxies for human evaluation. Table 5 lists the performances of different architectures and input lengths. The proposed Mamba model outperformed the Conformer model even with the same input length, and its performance was further improved by leveraging a longer input. While the Mamba model resulted in slightly worse performance compared to another sub-quadratic encoder known as FNet [52], this result indicates the potential of Mamba in long-form speech processing tasks.

7. Conclusion

In this paper, we explored the capability of Mamba in various speech applications, including ASR, TTS, SLU, and SUMM. The experimental results demonstrated that Mamba achieved a performance comparable to that of state-of-the-art Transformer variants across a wide range of benchmarks. In particular, Mamba exhibited advantages in long-form ASR and SUMM, not only in recognition performance but also in robustness and memory efficiency. We plan to conduct a detailed analysis focusing on the differences in behavior between Mamba and Transformer variants in future work.

8. References

[1] R. Prabhavalkar, T. Hori *et al.*, "End-to-end speech recognition: A survey," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 32, pp. 325–351, 2024.

[2] R. J. Weiss, J. Chorowski *et al.*, "Sequence-to-sequence models can directly translate foreign speech," in *Proc. Interspeech*, 2017, pp. 2625–2629.

[3] L. Barrault, Y.-A. Chung *et al.*, "Seamless: Multilingual expressive and streaming speech translation," *arXiv preprint arXiv:2312.05187*, 2023.

[4] A. Vaswani, N. Shazeer *et al.*, "Attention is all you need," in *Proc. NeurIPS*, 2017.

[5] S. Karita, N. Chen *et al.*, "A comparative study on Transformer vs RNN in speech applications," in *Proc. ASRU*, 2019, pp. 449–456.

[6] A. Gulati, J. Qin *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, 2020, pp. 5036–5040.

[7] K. Kim, F. Wu *et al.*, "E-Branchformer: Branchformer with enhanced merging for speech recognition," in *Proc. SLT*, 2022, pp. 84–91.

[8] Z. Yao, L. Guo *et al.*, "Zipformer: A faster and better encoder for automatic speech recognition," in *Proc. ICLR*, 2024.

[9] Y. Peng, K. Kim *et al.*, "A Comparative Study on E-Branchformer vs Conformer in Speech Recognition, Translation, and Understanding Tasks," in *Proc. Interspeech*, 2023, pp. 2208–2212.

[10] Y. Peng *et al.*, "Branchformer: Parallel MLP-attention architectures to capture local and global context for speech recognition and understanding," in *Proc. ICML*, 2022.

[11] T. Parcollet, R. van Dalen *et al.*, "SummaryMixing: A linear-complexity alternative to self-attention for speech recognition and understanding," 2023, arXiv:2307.07421.

[12] G. Saon, A. Gupta *et al.*, "Diagonal state space augmented transformers for speech recognition," in *Proc. ICASSP*, 2023, pp. 1–5.

[13] K. Miyazaki, M. Murata *et al.*, "Structured state space decoder for speech recognition and synthesis," in *Proc. ICASSP*, 2023, pp. 1–5.

[14] Y. Fathullah, C. Wu *et al.*, "Multi-head state space model for speech recognition," in *Proc. Interspeech*, 2023, pp. 241–245.

[15] H. Shan, A. Gu *et al.*, "Augmenting conformers with structured state-space sequence models for online speech recognition," in *Proc. ICASSP*, 2024, pp. 12 221–12 225.

[16] K. Goel, A. Gu *et al.*, "It's raw! audio generation with state-space models," in *Proc. ICML*, 2022, pp. 7616–7633.

[17] C. Chen, C.-H. H. Yang *et al.*, "A neural state-space modeling approach to efficient speech separation," in *Proc. Interspeech*, 2023, pp. 3784–3788.

[18] P.-J. Ku, C.-H. H. Yang *et al.*, "A multi-dimensional deep structured state space approach to speech enhancement using small-footprint models," in *Proc. Interspeech*, 2023, pp. 2453–2457.

[19] A. Gu, K. Goel *et al.*, "Efficiently modeling long sequences with structured state spaces," in *Proc. ICLR*, 2022.

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

[21] J. Wang, T. Gangavarapu *et al.*, "MambaByte: Token-free selective state space model," *arXiv preprint arXiv:2401.13660*, 2024.

[22] L. Zhu, B. Liao *et al.*, "Vision Mamba: Efficient visual representation learning with bidirectional state space model," *arXiv preprint arXiv:2401.09417*, 2024.

[23] A. Gu, K. Goel *et al.*, "On the parameterization and initialization of diagonal state space models," *Proc. NeurIPS*, vol. 35, pp. 35 971–35 983, 2022.

[24] J. T. Smith, A. Warrington *et al.*, "Simplified state space layers for sequence modeling," in *Proc. ICLR*, 2023.

[25] G. E. Blelloch, "Prefix sums and their applications," 1990.

[26] J. L. Ba, J. R. Kiros *et al.*, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[27] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997.

[28] H. Bu, J. Du *et al.*, "AISHELL-1: An open-source Mandarin speech corpus and a speech recognition baseline," in *Proc. O-COCOSDA*, 2017, pp. 1–5.

[29] G. Chen, S. Chai *et al.*, "GigaSpeech: An evolving, multi-domain ASR corpus with 10,000 hours of transcribed audio," in *Proc. Interspeech*, 2021, pp. 4376–4380.

[30] K. Maekawa, H. Koiso *et al.*, "Spontaneous speech corpus of Japanese," in *Proc. LREC*, 2000.

[31] V. Panayotov, G. Chen *et al.*, "LibriSpeech: An ASR corpus based on public domain audio books," in *Proc. ICASSP*, 2015.

[32] A. Rousseau, P. Deléglise *et al.*, "Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks." in *Proc. LREC*, 2014, pp. 3935–3939.

[33] "VoxForge." [Online]. Available: http://www.voxforge.org/

[34] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. ICLR*, 2019.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[36] K. Ito and L. Johnson, "The LJ speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[37] T. Hayashi, R. Yamamoto *et al.*, "ESPnet2-TTS: Extending the edge of tts research," *arXiv preprint arXiv:2110.07840*, 2021.

[38] Y. Ren, C. Hu *et al.*, "FastSpeech 2: Fast and high-quality end-to-end text to speech," in *Proc. ICLR*, 2020.

[39] J. Kong, J. Kim *et al.*, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," *Proc. NeurIPS*, vol. 33, pp. 17 022–17 033, 2020.

[40] M. McAuliffe, M. Socolof *et al.*, "Montreal forced aligner: Trainable text-speech alignment using Kaldi," in *Proc. Interspeech*, 2017, pp. 498–502.

[41] E. Bastianelli, A. Vanzo *et al.*, "SLURP: A spoken language understanding resource package," in *Proc. EMNLP*, 2020.

[42] S. Shon, A. Pasad *et al.*, "SLUE: New benchmark tasks for spoken language understanding evaluation on natural speech," in *Proc. ICASSP*, 2022.

[43] S. Arora, S. Dalmia *et al.*, "ESPnet-SLU: Advancing spoken language understanding through ESPnet," in *Proc. ICASSP*, 2022.

[44] Y. Peng, S. Arora *et al.*, "A study on the integration of pre-trained ssl, asr, lm and slu models for spoken language understanding," in *Proc. SLT*, 2022, pp. 406–413.

[45] R. Sharma, S. Palaskar *et al.*, "End-to-end speech summarization using restricted self-attention," in *Proc. ICASSP*, 2022, pp. 8072–8076.

[46] T. Kano, A. Ogawa *et al.*, "Speech summarization of long spoken document: Improving memory efficiency of speech/text encoders," in *Proc. ICASSP*, 2023, pp. 1–5.

[47] R. Sanabria, O. Caglayan *et al.*, "How2: a large-scale dataset for multimodal language understanding," in *Proc. ViGIL*, 2018.

[48] R. Sharma, W. Chen *et al.*, "Espnet-Summ: Introducing a novel large dataset, toolkit, and a cross-corpora evaluation of speech summarization systems," in *Proc. ASRU*, 2023, pp. 1–8.

[49] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proc. Summarization Branches Out*, 2004, pp. 74–81.

[50] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proc. ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 2005, pp. 65–72.

[51] T. Zhang, V. Kishore *et al.*, "BERTScore: Evaluating text generation with BERT," in *Proc. ICLR*, 2019.

[52] J. Lee-Thorp, J. Ainslie *et al.*, "FNet: Mixing tokens with Fourier transforms," in *Proc. NAACL*, 2022, pp. 4296–4313.