

ConMamba: A Convolution-Augmented Mamba Encoder Model for Efficient End-to-End ASR Systems

Haoxiang Hou, Xun Gong, Yanmin Qian[†]

Auditory Cognition and Computational Acoustics Lab

MoE Key Lab of Artificial Intelligence, AI Institute

Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai

houhaoxiang0701@sjtu.edu.cn, gongxun@sjtu.edu.cn, yanminqian@sjtu.edu.cn

Abstract

End-to-End Automatic Speech Recognition (ASR) models, such as Conformer, excel in accuracy but face limitations in computational complexity and positional awareness, hindering their use in real-time or resource-constrained settings. State Space Models (SSMs), particularly the Mamba model with its time-varying mechanism, offer a more efficient alternative. We propose ConMamba, a Convolution-Augmented Mamba Encoder model, which replaces the Conformer's Multi-Head Self-Attention with Mamba layers and adds convolutional layers for capturing both local and global features. Experiments on the LibriSpeech dataset show ConMamba matches traditional Conformer performance on short speech segments and outperforms them on longer ones, enhancing robustness and efficiency for practical ASR applications.

Index Terms: speech recognition, state space model

1. Introduction

The End-to-End Automatic Speech Recognition (ASR) model built on the Transformer architecture has demonstrated impressive performance across various speech recognition tasks. This model leverages the strengths of the Transformer's self-attention mechanism, which allows it to capture long-range dependencies and complex patterns within speech data. The ability of the Transformer to process entire sequences of input data simultaneously, rather than incrementally, contributes to its high accuracy. Consequently, Transformer-based ASR models have become a benchmark for performance in the field of speech recognition, often surpassing traditional models in accuracy [1, 2, 3].

Despite its high performance, the Transformer model suffers from a significant drawback: its computational complexity increases quadratically with the size of the context window. This limitation becomes particularly problematic in real-time or resource-constrained environments, where the available computational power and memory are limited. Consequently, the quadratic growth in complexity can hinder the practical deployment of Transformer-based ASR models in such scenarios, necessitating the search for more efficient alternatives that can handle large context windows without overwhelming computational demands.

State Space Models (SSMs), such as the Structured State Space Sequence model (S4), offer a promising alternative for ASR tasks due to their efficient computational complexity and inherent positional awareness. SSMs are designed to handle sequential data by maintaining a state that evolves over time,

which naturally incorporates positional information. This design allows SSMs to model temporal dependencies more effectively than Transformers. Moreover, the computational complexity of SSMs typically scales linearly with the size of the input, making them much more efficient than Transformers for processing long sequences. These attributes make SSMs particularly well-suited for ASR tasks, where capturing the temporal structure of speech is crucial for accurate recognition [4, 5, 6, 7].

Mamba, a newly developed model that incorporates a time-varying mechanism into State Space Models (SSMs), has shown exceptional performance across multiple domains, including computer vision (CV), natural language processing (NLP), and speech processing. In CV, Mamba effectively captures spatial and temporal dynamics, leading to superior image and video understanding. In NLP, Mamba excels at modeling complex linguistic structures and dependencies, enhancing tasks like language modeling and translation. For speech processing, Mamba's ability to integrate time-varying mechanisms allows it to handle the intricacies of spoken language with high accuracy. Its versatility and robust performance across these diverse fields underscore Mamba's potential as a powerful tool in modern AI applications [8, 9, 10, 11].

In this work, we introduce the Mamba model into end-to-end Automatic Speech Recognition (ASR) systems as a replacement for the Multi-Head Self-Attention (MHSA) mechanism, offering significant advancements. Mamba maintains comparable performance to traditional Transformer-based models like Conformer for short speech segments while addressing computational complexity and positional awareness issues. Our proposed Convolution-Augmented Mamba (ConMamba) Encoder model leverages both convolutional and state-space mechanisms, capturing local and global features more effectively. This combination results in a model that not only matches the performance of existing ASR systems for short speech segments but also surpasses them in handling longer speech segments. This enhanced capability significantly improves overall robustness and efficiency in speech recognition tasks, making it a viable solution for real-world ASR applications. Experiments on the LibriSpeech dataset show that ConMamba significantly outperforms existing models, reducing error rates from 71.0% to 32.3% for 60-second audio, highlighting its potential for real-world ASR applications [12, 13, 14].

2. Related Work

2.1. End-to-end Automatic Speech Recognition Models

End-to-end Automatic Speech Recognition (ASR) models can be classified into three primary types based on their training objectives and decoding mechanisms. The first type is Connec-

[†]: Corresponding Author

tionist Temporal Classification (CTC) models [15], which address the alignment problem between input speech frames and output transcription sequences without requiring pre-segmented training data. The second type is Transducer models[16], which extend the capabilities of CTC by integrating an additional prediction network. The third type is the Attention-Based Encoder-Decoder (AED) models, also known as Listen, Attend, and Spell (LAS) models[17]. AED models have shown remarkable performance in capturing long-range dependencies and contextual information, making them a popular choice for various ASR tasks.

The Attention-Based Encoder-Decoder (AED) model employs a sophisticated architecture that integrates convolutional and attention mechanisms within its encoder. This capability is crucial for modeling long-term dependencies and contextual relationships, which are essential for accurate speech recognition [18, 19, 20]. The combination of convolutional and attention mechanisms within the encoder enables the AED model to effectively balance the extraction of both local and global features, leading to enhanced performance.

2.2. State Space Models

State Space Models (SSMs) originate from continuous systems and serve as a powerful mathematical framework for modeling dynamic systems. In continuous systems, the state of the system evolves over time according to a set of differential equations 1. These equations describe the relationship between the input $x(t) \in \mathbb{R}$, the current state $h(t)$ in $\mathbb{R}^n$, and the output $y(t) \in \mathbb{R}$ of the system. $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times d}$, $C \in \mathbb{R}^{d \times n}$, $D \in \mathbb{R}^{d \times d}$ are parameters of the model. The fundamental equations are represented as:

$$h'(t) = Ah(t) + Bx(t), \quad y(t) = Ch(t) + Dx(t). \quad (1)$$

To apply State Space Models in digital systems, it is necessary to transform the continuous representation into a discrete form. This transformation is commonly achieved using the Zero-Order Hold (ZOH) method as 2, where Δ is a scaling parameter meaning sample time interval, $\tilde{A}$ and $\tilde{B}$ is the new parameters of discrete equations.

$$\tilde{A} = \exp \Delta A, \quad \tilde{B} = (\Delta A)^{-1}(\exp \Delta A - I)\Delta B. \quad (2)$$

Then the dynamic equations are represented as 3. $D \in \mathbb{R}^{d \times d}$ can be considered as the residual connection in Neural Network.

$$h_t = \tilde{A}h_{t-1} + \tilde{B}x_t, \quad y_t = Ch_t. \quad (3)$$

In the context of traditional State Space Models, the assumption of Linear Time Invariance (LTI) plays a crucial role in simplifying the analysis and implementation of the model. Under the LTI assumption, the system's dynamic parameters, A, B, C , remain constant over time. The LTI assumption allows for the use of powerful mathematical tools and techniques, such as convolution as follow equation to compute.

$$y = x * \tilde{K}, \quad \tilde{K} = (C\tilde{B}, C\tilde{A}\tilde{B}, \dots, C\tilde{A}^{L-1}\tilde{B}) \quad (4)$$

The Mamba model [8], an advancement over traditional SSMs, introduces significant enhancements by removing the Linear Time Invariance (LTI) constraint and incorporating a selection mechanism. By relaxing the LTI constraint, Mamba introduces a selection mechanism that allows the dynamic parameters of the system to vary over time, enabling the model to

adapt to non-stationary environments and capture more complex behaviors. This mechanism is particularly beneficial for tasks that involve long-range dependencies and varying contextual importance.

3. Method

3.1. Mamba Encoder

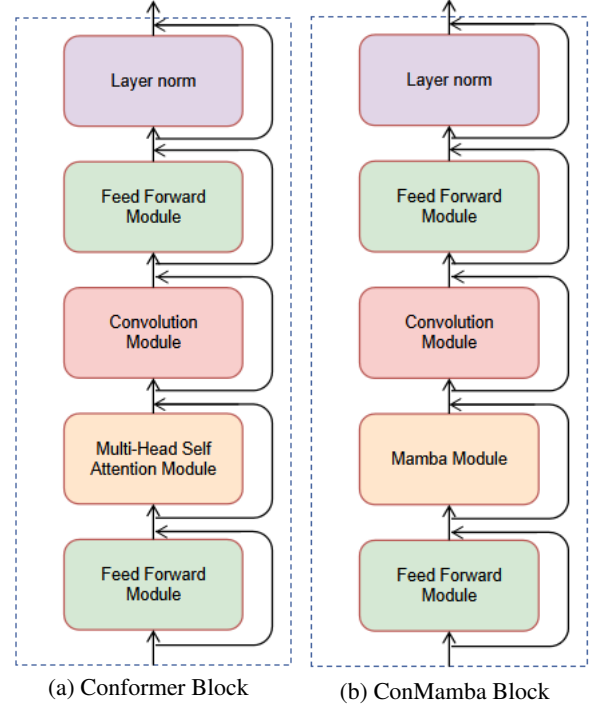


Figure 1: **Conformer and ConMamba block model architecture.** ConMamba is inspired by the Conformer architecture, but it replaces the Multi-Head Self-Attention (MHSA) with the Mamba module.

In the realm of computer vision, many attempts to leverage the Mamba architecture involve directly substituting traditional transformer layers with Mamba modules [9, 10]. In line with this approach, we utilize Mamba layers as a direct replacement for transformer layers to construct the Mamba encoder model.

Our methodology follows the common attention-based encoder-decoder (AED) model framework, which is widely used in ASR tasks. The architecture begins with a convolutional subsampling layer to reduce the input sequence length while preserving essential information. This subsampled input is then fed into a series of Mamba blocks. Each Mamba block is a simplified State Space Model architecture, as detailed by Gu et al. [8]. The decoder in our model is a conventional transformer decoder, which allows for effective sequence generation by attending to the encoder's outputs.

The training of the model is carried out using a joint loss function that combines Connectionist Temporal Classification (CTC) and attention loss. This combination leverages the strengths of both loss functions: CTC's capability to handle unsegmented data and attention loss's ability to align and generate sequences more accurately.

3.2. Convolution-Augmented Mamba Encoder

For audio processing tasks, it is essential to extract both global and local hidden features to achieve high performance. Previous research, such as Conformer [21] and Branchformer [3], has demonstrated the effectiveness of combining convolutional layers with self-attention mechanisms to achieve this goal. Conformer, for instance, utilizes convolutional layers to capture local hidden features and self-attention mechanisms to capture global hidden features, achieving a robust representation of the input data.

Building on these insights, we propose a Convolution-Augmented Mamba Encoder model. In this model, we replace the multi-head self-attention mechanism in the Conformer encoder with Mamba layers. The convolutional layers are retained to enhance the Mamba layers with the capability to capture local features. This hybrid approach allows the model to effectively extract and integrate both local and global features, improving overall performance.

The Convolution-Augmented Mamba Encoder model retains the Macaron sandwich structure of the Conformer, which consists of two feed-forward modules. This structure ensures a balanced representation by processing the input through multiple paths.

Mathematically, for an input x_i to a ConMamba block i , the output y_i is computed as follows:

$$\begin{aligned}\tilde{x}_i &= x_i + \frac{1}{2}\text{FFN}(x_i) \\ x'_i &= \tilde{x}_i + \text{Mamba}(\tilde{x}_i) \\ x''_i &= x'_i + \text{Conv}(x'_i) \\ y_i &= \text{Layernorm}(x''_i + \frac{1}{2}\text{FFN}(x''_i))\end{aligned}\quad (5)$$

In this context, FFN denotes the feed-forward module, Mamba refers to the Linear-Time Sequence Model with Selective State Spaces, and Conv represents the convolution module. The process begins by adding half of the output of the feed-forward network (FFN) to the input x_i to form $\tilde{x}_i$. This is followed by the application of the Mamba layer to $\tilde{x}_i$, resulting in x'_i . Next, the convolutional layer processes x'_i to produce x''_i . Finally, layer normalization is applied to x''_i combined with half of the output from another FFN to generate the final output y_i .

Combining local and global features has been extensively studied, and there are various methods to achieve this integration. Our experimental results indicate that augmenting Mamba layers with a convolution module significantly improves performance compared to using Mamba layers alone in the encoder. This improvement can be attributed to the convolutional layers' ability to capture fine-grained local patterns, which complements the Mamba layers' capacity to model long-range dependencies. By effectively integrating these features, the Convolution-Augmented Mamba Encoder achieves superior performance in audio processing tasks.

4. Experiment

4.1. Dataset

We evaluate our models using the LibriSpeech-100 dataset [22], which consists of recordings sampled at 16 kHz. LibriSpeech-960, a widely utilized dataset in ASR, contains approximately 1000 hours of audio recordings and their corresponding transcriptions. The LibriSpeech dataset is divided into three training

subsets: LibriSpeech-100, LibriSpeech-360, and LibriSpeech-500. We choose LibriSpeech-100 for its superior recording quality. The ASR performance on LibriSpeech-100 is assessed using both clean and noisy development and test sets, labeled as dev-clean, dev-other, test-clean, and test-other.

To verify robustness on long-form input, we create a long-form dataset by concatenating short-form audio files from LibriSpeech. We train the model on the original short-form LibriSpeech-100 training set and then test it on the long-form concatenated data. A threshold is employed to control the length of the concatenated data. Short-form data are added incrementally until the accumulated length exceeds the threshold, at which point they are concatenated into a long-form dataset.

4.2. Experimental Setup

All experiments were conducted using the ESPnet toolkit, with the baseline being an AED model featuring a Conformer encoder and Transformer decoder. For the Mamba encoder, we doubled the number of layers compared to the Conformer encoder to match the parameter size. The ConMamba encoder retained the same hyperparameters as the baseline Conformer encoder. All models used a Transformer decoder, without language models or speed perturbation.

We evaluated two models with parameter sizes of 100M and 30M. Our configuration and preprocessing followed the LibriSpeech Conformer recipe. For the 100M models, the encoder had 12 layers with an output size of 512, 8 attention heads, 2048 linear units, and dropout rates of 0.1. The decoder had 6 layers, each with 8 attention heads and 2048 linear units, and a dropout rate of 0.1. The model configuration included a CTC weight of 0.3. Additional settings involved sorting in batches and accumulating gradients over 4 steps. Training lasted up to 70 epochs, using Adam as the optimizer with a learning rate of 0.0025 and a weight decay of 0.000001, with a learning rate scheduler configured for a 40,000-step warmup period. For the 30M models, the encoder had an output size of 256, 4 attention heads, and 1024 linear units. Each decoder layer had 4 attention heads and 2048 linear units, with a peak learning rate of 0.002. For both the Mamba and ConMamba encoders, the Mamba layer dimension matched the encoder output size, with an SSM state expansion factor of 16, a local convolution width of 4, and a block expansion factor of 2.

4.3. Results on LibriSpeech Clean 100

Table 1 and Table 2 present the Word Error Rate (WER) results for both the baseline and our models on the LibriSpeech clean 100 dataset. In our experiments, the ConMamba model configured with a real number state achieves results comparable to those of the Conformer model, demonstrating its efficacy in maintaining high performance.

On the other hand, the ConMamba model configured with a complex number state shows a higher WER compared to the real number state. This outcome suggests that the inclusion of complex numbers in the Mamba state does not contribute positively to the model's performance and is thus unnecessary. The increased error rate indicates that the complexity introduced by using complex numbers does not translate into better model accuracy.

Furthermore, the performance of the Mamba encoder alone is found to be weaker than that of the ConMamba model. This disparity highlights the importance of the convolution module present in the ConMamba architecture. The convolution module appears to be crucial for effective local feature extraction,

Table 1: WER[%] results on LibriSpeech clean 100. The Conformer model is configured as specified in ESPnet. The ConMamba model replaces the Multi-Head Self-Attention (MHSA) layer in the Conformer with the Mamba layer. Mamba uses Mamba layers as a direct replacement for transformer layers. Note that the experiments were conducted without speed perturbation. #Params(M) indicates the number of parameters in millions.

Method	#Params(M)	WER[%](↓)			
		dev clean	dev other	test clean	test other
Conformer	116.15	7.1	18.2	7.4	18.3
ConMamba	120.72	7.3	20.2	7.6	20.6
Mamba	123.89	19.0	40.4	19.2	42.4

Table 2: WER[%] results on LibriSpeech clean 100. The Conformer model is configured as specified in ESPnet. The ConMamba model replaces the Multi-Head Self-Attention (MHSA) layer in the Conformer with the Mamba layer. ConMamba (Real) uses real numbers in its state, while ConMamba (Complex) uses complex numbers. Mamba uses Mamba layers as a direct replacement for transformer layers. The experiments were conducted with speed perturbation. #Params(M) indicates the number of parameters in millions.

Method	#Params(M)	WER[%](↓)			
		dev clean	dev other	test clean	test other
Conformer	34.23	6.5	17.5	6.8	17.6
ConMamba(Real)	35.53	6.7	18.5	7.0	19.1
ConMamba(Complex)	35.53	7.0	18.9	7.1	19.2
Mamba	34.40	9.5	24.5	9.7	25.2

which significantly enhances the model’s ability to process and interpret audio data accurately. The results suggest that the ConMamba model benefits from the combination of the Mamba layer and the convolution module, leading to a more robust and precise speech recognition system.

In summary, while the ConMamba model with a real number state is highly effective and on par with the Conformer model, the use of complex numbers in the Mamba state is unnecessary and even detrimental. Additionally, the convolution module’s presence in ConMamba is vital for achieving superior performance, underscoring its role in local feature extraction.

4.4. Robustness on Long-form Input

We conducted tests on our model using long-form audio data. This evaluation is particularly meaningful because, in real-world applications, audio may not always be well-segmented. Our model was initially trained on the raw, short dataset LibriSpeech clean 100. To understand the robustness and performance of our model, we then tested it on long speech audio data, specifically focusing on audio lengths that the model did not encounter during training.

Table 3 and Fig 2 present the WER results for LibriSpeech across various audio lengths. The raw short dev-clean dataset typically includes segments of approximately 10 seconds. For the raw dataset and the dataset with a 20-second threshold, the ConMamba model shows performance comparable to the Conformer model. However, when the audio length threshold exceeds 30 seconds, the ConMamba model significantly outper-

Table 3: WER[%] results on LibriSpeech dev-clean. "Raw" refers to the original short dev-clean data, while the others are concatenated based on the time threshold length. #Params(M) indicates the number of parameters in millions.

Method	#Params(M)	dev-clean WER[%](↓)				
		raw	20s	30s	40s	60s
Conformer	116.15	7.1	7.1	11.0	39.6	71.0
ConMamba	120.72	7.3	7.4	8.9	15.6	32.3

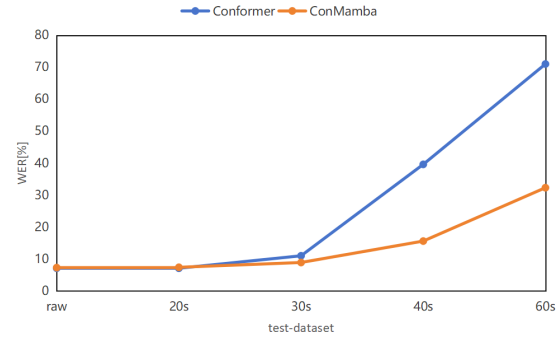


Figure 2: WER on LibriSpeech distribution by test dataset with different speech length.

forms the Conformer model. This improvement suggests that ConMamba is substantially more robust than the Conformer baseline.

The enhanced robustness of ConMamba in handling longer audio segments can be attributed to the reduced complexity of the State Space Model (SSM) it employs. By simplifying the model, ConMamba manages to maintain performance consistency across varying audio lengths, a critical factor in real-world audio processing where segment lengths are often unpredictable. This capability makes ConMamba a more reliable choice for applications involving continuous or lengthy audio streams.

5. Conclusion

In this work, we introduce ConMamba, an innovative model for end-to-end automatic speech recognition (ASR). ConMamba integrates the Mamba model into ASR frameworks, overcoming the high computational complexity and lack of positional information in traditional Transformer-based models. This Convolution-Augmented Mamba Encoder combines convolutional and state space mechanisms, offering robust and efficient performance for both short and long speech segments. Empirical evaluations on the LibriSpeech dataset demonstrate ConMamba’s superiority, particularly with long-form audio, achieving a performance improvement from 71.0% to 32.3% for 60-second audio, highlighting its potential for real-world ASR applications and more accurate speech recognition systems.

6. Acknowledgements

This work was supported in part by China NSFC projects under Grants 62122050 and 62071288, in part by Shanghai Municipal Science and Technology Commission Project under Grant 2021SHZDZX0102.

7. References

- [1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NeurIPS*, 2017.
- [2] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 5036–5040.
- [3] 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 *ICML*, 2022.
- [4] A. Gu, K. Goel, and C. Re, "Efficiently modeling long sequences with structured state spaces," in *International Conference on Learning Representations*, 2021.
- [5] 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," *Advances in neural information processing systems*, vol. 34, pp. 572–585, 2021.
- [6] A. Gu, K. Goel, A. Gupta, and C. Ré, "On the parameterization and initialization of diagonal state space models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 35 971–35 983, 2022.
- [7] A. Gupta, A. Gu, and J. Berant, "Diagonal state spaces are as effective as structured state spaces," *Advances in Neural Information Processing Systems*, vol. 35, pp. 22 982–22 994, 2022.
- [8] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.
- [9] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, "Vision mamba: Efficient visual representation learning with bidirectional state space model," *arXiv preprint arXiv:2401.09417*, 2024.
- [10] Y. Yang, Z. Xing, and L. Zhu, "Vivim: a video vision mamba for medical video object segmentation," *arXiv preprint arXiv:2401.14168*, 2024.
- [11] O. Lieber, B. Lenz, H. Bata, G. Cohen, J. Osin, I. Dalmedigos, E. Safahi, S. Meirum, Y. Belinkov, S. Shalev-Shwartz *et al.*, "Jamba: A hybrid transformer-mamba language model," *arXiv preprint arXiv:2403.19887*, 2024.
- [12] X. Jiang, C. Han, and N. Mesgarani, "Dual-path mamba: Short and long-term bidirectional selective structured state space models for speech separation," *arXiv preprint arXiv:2403.18257*, 2024.
- [13] K. Li and G. Chen, "Spmamba: State-space model is all you need in speech separation," *arXiv preprint arXiv:2404.02063*, 2024.
- [14] C. Quan and X. Li, "Multichannel long-term streaming neural speech enhancement for static and moving speakers," *arXiv preprint arXiv:2403.07675*, 2024.
- [15] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006.
- [16] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.
- [17] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.
- [18] 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," in *ICASSP*, 2020.
- [19] L. Dong, S. Xu, and B. Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *ICASSP*, 2018.
- [20] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang *et al.*, "A comparative study on transformer vs rnn in speech applications," in *ASRU*, 2019.
- [21] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *Interspeech*, 2020.
- [22] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*, 2015.