INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Conmer: Streaming Conformer without self-attention for interactive voice assistants

Martin Radfar, Paulina Lyskawa, Brandon Trujillo,Yi Xie, Kai Zhen, Jahn Heymann, Denis Filimonov, Grant Strimel, Nathan Susanj, Athanasios Mouchtaris

Alexa Machine Learning, Amazon, USA
{radfarmr,lyskawpa,btruj,yixiey,kaizhen,
jahheyma,denf,gsstrime,nsusanj,mouchta}@amazon.com

Abstract

Conformer is an extension of transformer-based neural ASR models whose fundamental component is the self-attention module. In this paper, we show that we can remove the self-attention module from Conformer and achieve the same or even better recognition performance for utterances whose length is up to around 10 seconds. This is particularly important for streaming interactive voice assistants as input is often very short and a fast response is expected. Since the computational complexity of self-attention is quadratic, this modification allows for faster, smaller sized models, two requirements for on-device applications. Using this finding, we propose Conmer, a neural architecture based on Conformer but without self-attention for streaming interactive voice assistants. We conduct experiments on public and real-world data and show the streaming Conmer reduces the WER and computational complexity relatively by 4.03% and 10%, respectively.

Index Terms: Conformer, self-attention, Convolutional neural networks, sequence-to-sequence, Transformer

1. Introduction

The field of automatic speech recognition (ASR) has gone through a fast transition from LSTM-based neural architectures to attention-based architectures in the past four years [1–8]. As conversational AI becomes widespread, demand for streaming and real-time ASR has increased remarkably where fast response times are crucial for the user experience, such as virtual assistants or live video stream caption generation. Sequence-to-Sequence (Seq2Seq) neural-based ASR have been deployed in many voice assistant devices as they are streamable, accurate, and low-footprint [9–13]. A class of Seq2Seq ASR models are transducers which consist of an audio encoder, a label encoder and a joint network based on seminal work known as recurrent neural network transducers (RNN-T) [9]. Before the introduction of Transformers [14], both audio and label encoders of ASR transducers are comprised of stacks of LSTM layers which have been dominant neural modules for sequence modeling.

Transformer-Transducers were introduced as a replacement for LSTM-based transducers where the encoder of RNN-T is replaced with a Transformer encoder [2, 3]. In order to make these models streamable, a causal self-attention mechanism is deployed where the current frame is only allowed to attend to left frames to make the model real-time [2–5, 7, 8]. In this design, ASR emits predictions for each frame as they arrive from the audio signal, without access to future frames.

Pioneering Transformer Transducers had difficulty to deliver state-of-the art results [2,3]. Accordingly, several improvements have been made to make them compatible with audio signals. First, it is shown that the use of a convolutional frontend is necessary to capture the local audio information [2, 15]. Consequently, a CNN frontend has become a standard module in Transformer Transducers encoders. In order to further improve the performance of Transformer Transducer and drawing inspiration from convolutional ASR models [16–18], Conformer was proposed where the convolutional blocks are not only used at the frontend but also inserted between multi-head self-attention blocks and feedforward networks in the encoder block [19]. Introducing convolutional blocks into Transformer Transducer improves the performance substantially and makes Conformer one of the state-of-the-art E2E ASR models.

The Conformer architecture consists of self-attention, feedforward, and convolutional blocks stacked in an elegant neural topology. Because self-attention is considered a fundamental block in Transformer Transducers, previous research has only considered the incremental role of feedforward and CNN layers in improving accuracy for the Conformer. For instance, [19] studies the impact of removing all modules on performance except for self-attention. Despite this added benefit, the following question remains unanswered: what is the performance of Conformer if we remove the self-attention block? This question is particularly important as self-attention is computationally expensive, which is a bottleneck for real-time as well as on-device voice assistant systems. Moreover, for causal ASR, self-attention loses its power to provide global context for better prediction because it can only be computed on left frames. Recent studies also show that the heatmap of attention heavily concentrates on the diagonal, meaning attention mostly learns local context, especially for upper layers [20]. In addition, the self-attention block has been shown to consume excessive power to operate [21, 22]. Finally, on-device ASR applications are often executed on neural hardware accelerators where implementing self-attention has been shown to be challenging and costly [23].

In this paper, we remove self-attention from streaming (causal) Conformer and introduce a simpler architecture named Conmer. We show that removing self-attention does not impact the performance of Conformer for short utterances. Our design is particularly important for streaming interactive voice assistants where communications mostly comprise of short transactional utterances. To confirm these design elements, we benchmark the performance of Conformer and Conmer with utterances of different length ranges. We also investigate the impact of self-attention for non-streaming (non-causal) Conformer with and without self-attention. Our results show the importance of future context for increasing the impact of self-attention on prediction accuracy. Because Conmer only consists of feedforward and CNN modules, it is parallelizable, streamable, computationally less expensive, more hardware and quan-

10.21437/Interspeech.2023-2228

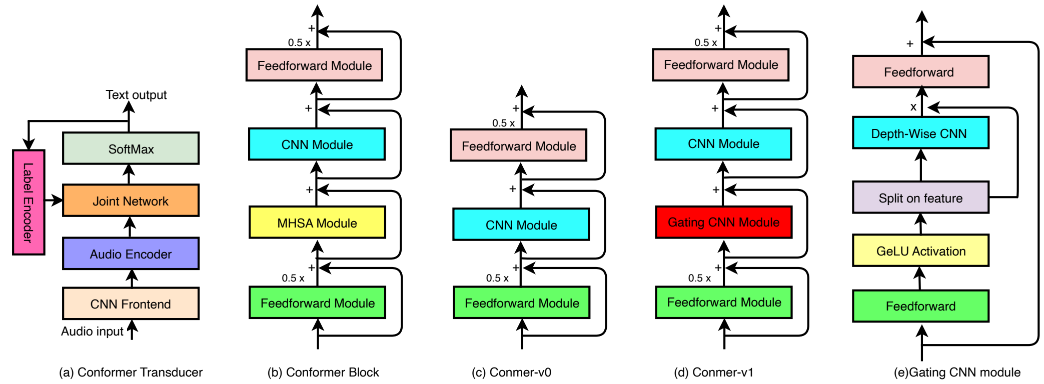

Figure 1: *The high-level block-diagram of Conformer and Conmer: (a) Conformer transducer, (b) Conformer block, (c) Conmer-v0, (d) Conmer-v1, and (e) the gating depth-wise CNN module used in Conmer-v1. For simplicity, we omitted layer normalization and dropouts from the block-diagrams.*

tization friendly, and yet delivers similar accuracy to the Conformer. These features make Conmer a promising candidate for on-device streaming interactive voice assistants.

Furthermore, our work aligns with recent proposals to replace attention with multi-layer perceptron and gating modules in several research areas [24]. In particular, [24] proposes a model without attention that can achieve the same performance of Transformer for natural language processing applications. Their approach, however, cannot be directly applied to variable sequence prediction such as speech. As such, [25] modified work done in [24] to apply it for ASR and reported encouraging results. In addition, modern, all-CNN-based ASR models have been shown to deliver the state-of-the-art results without attention and feedforward neural modules [16–18]. These models however require stacking many layers (e.g. > 100), which introduces latency. Moreover, gradient vanishing becomes an issue for training models with excessively large numbers of layers. Our Conmer approach not only takes advantage of the Conformer topology but is simpler, faster, and easier to implement and train. In addition, our design is more quantization-friendly which is an important mechanism to leverage all these architectures for on-device usage.

2. Conmer architecture

In the following subsections, we briefly describe the Conformer transducer and introduce two variants of Conmer architectures.

2.1. Conformer transducer architecture

The Conformer design is based on RNN transducers [9, 12] and is composed of the CNN frontend, audio and label encoders, joint network, and a Softmax layer as depicted in Figure 1-a and detailed in [19]. For the sake of brevity, we omitted layer normalization and dropouts. The input audio sequence is transferred to a time-frequency space represented by $\boldsymbol{x} = (x_1, \ldots, x_i, \ldots, x_T)$ where $x_i \in \mathcal{R}^D$, T and D denote the number of frames and the dimension of the frequency space, respectively. The CNN frontend down-samples the input sequence to the length equal to $T' = \frac{1}{4}T$. The audio encoder transforms the sequence to higher level representations denoted by $\boldsymbol{h}^L = (h_1^L, \ldots, h_i^L, \ldots, h_{T'}^L)$, where L is the number of layers of the audio encoder. The Conformer audio encoder consists of four main blocks: two feedforward modules (FFM), a multi-head self-attention, and a CNN-based module. Each of these modules has a residual connection which has been shown to alleviate vanishing gradient and speed up convergence [26]. The FFM module consists of two cascaded linear projections where the output of the first one is passed through the Swish activation function which has been reported to give better results than ReLu. The output of each FFM is weighted by half. The weighted output of FFM is added to the input and passed to multi-head self-attention (MHSA). MHSA computes the weighted sum of all frames (for causal case only left frames) for the current frame in a lower dimension feature space. The weights (the so-called attention coefficients) are obtained by a dot product of two vectors called key and query followed by the Softmax operation and a division by the square root of the lower dimension.

The CNN module comprises of three cascaded CNN layers, a depth-wise CNN sandwiched between two point-wise CNNs. The point-wise CNN and the depth-wise CNN are followed by the gating linear unit (GLU) and a Swish activation, respectively and are preceded by another point-wise CNN. Using depth-wise and point-wise convolutions reduces the computational complexity and has shown to improve performance in large scale vision problems [27, 28] and ASR [16]. Finally, the output of the CNN module is passed to a mirror FFM to obtain higher level representation denoted by $\boldsymbol{h}^l = (h_1^l, \ldots, h_i^l, \ldots, h_{T'}^l)$ for the lth layer of the audio encoder.

2.2. Conmer

The Conmer architecture is adapted from Conformer. It is a Conformer transducer where we make strategic modifications in the Conformer block. All other components remain the same. Here, we introduce two versions of Conmer.

2.2.1. Conmer-v0

Conmer-v0 is simply the same as Conformer but without MHSA as illustrated in Figure 1-c. This architecture is attractive for applications in which fast inference is desired or deployed for low resource use cases such as on-device voice assistants. Our experiments show Conmer-v0 has less computational complexity than Conformer and delivers better predictive performance. Conmer is also attractive for real world applications where some hardware platforms do not support MHSA or encounter limitation due to power consumption.

2.2.2. Conmer-v1

Conmer-v1, illustrated in Figure 1-d is inspired by multi-layer perceptron with a gating mechanism originally proposed in [24] and extended to ASR in [25]. We build this model to investigate whether Conformer without self-attention benefits from the gating CNN unit (Figure 1-e), which is used in lieu of MHSA in previous works [24, 25].

3. Experiments

In the following subsections, we first describe the data and model parameters and next we report the results in terms of accuracy and complexity of models.

3.1. Data and model parameters

We use the Librispeech corpus which consists of 970 hours of labeled speech [29] and 50K hours of our de-identified in-house data to benchmark our models against Conformer; we split our evaluation in-house data into seven sets, each of which comes from different traffic. We used 64-dimensional log short time Fourier transform vectors obtained by segmenting the utterances with a Hamming window of the length 25 ms and frame shift of 10 ms. The three frames are stacked resulting in 192-dimensional input features.

First, we build a streaming Conformer transducer. The acoustic encoder has 14 layers in which we use multi-head attention with four heads and each head with a dimension of 64. We make Conformer causal (streamable) by applying masks to multi-head attention layers to only attend to left context as well as building all convolutional layers to be causal. For the convolutional sub-sampling block of Conformer, we use two layers of 2D CNN with filters of 128 channels, kernel size of three, and stride of two. The feedforward hidden unit dimension is set to 1,024.

In Conmer-v1, the dimension of the first feedforward is set to 1,024, while the second one reduces the dimension back to 256. For the depth-wise CNN, we use the kernel size of 32 and the stride of one. In Conmer-v1 after the GeLu activation, the tensor is split along the channel dimension with one half then undergoing depth-wise convolution before being multiplied with the other (non-transformed) half. We also built a baseline RNN-T of six LSTM layers and a hidden unit of size 640. The label encoder has one layer of unidirectional LSTM with 640 hidden units and dropout of 0.1, and we add L_2 regularization of $1e-6$ to all trainable weights. The label encoder remains the same for all models for which we report the results. The dimension of the encoder output is set to 256 for all models. We used greedy decoding and no language model is used. We add more regularization and robustness using SpecAug [30] with the following hyper-parameters: maximum ratio of masked time frames=0.04, adaptive multiplicity=0.04, maximum ratio of masked frequencies=0.34, and number of frequency masks=2. We use a word-piece tokenizer and generate 2,500 word-piece tokens as the output vocabulary. The number of parameters of each component of Conformer transducer is given in Table 2. We use the Adam optimizer with β_1=0.9, β_2= 0.98, and ϵ=1e-9. The learning curve was chosen to have high pick of 0.002 and warm-up rate of 10,000. We used step size of 5,000 for Librispeech and in-house data, and the model trained until no improvement was observed. The models were trained using three machines each of which has eight NVIDIA Tesla V100 GPUs.

Table 1: *WER (%) vs. utterance length for the test-other Librispeech corpus; T denotes the length of the utterance.*

	$T \leq 10$(sec)		$T > 10$(sec)	
	causal	non-causal	causal	non-causal
Conformer	21.21	15.96	**13.23**	9.13
Conmer-v0	20.79	18.22	14.95	12.97
Conmer-v0(d_{ff}=1,524)	**20.39**	17.89	14.07	12.93

Table 2: *WER(%) results on the Librispeech test for the causal models.*

Model	dev-clean	dev-other	test-clean	test-other	Size (M)
LSTM	5.63	14.94	5.96	14.74	30
Conformer	5.20	**13.22**	5.56	**13.32**	28
Conmer-v0	5.39	14.11	5.53	14.11	23
Conmer-v0 (d_{ff}=1,524)	**5.07**	13.70	**5.22**	13.78	29
Conmer-v1	5.25	13.74	5.66	13.56	28

3.2. Results

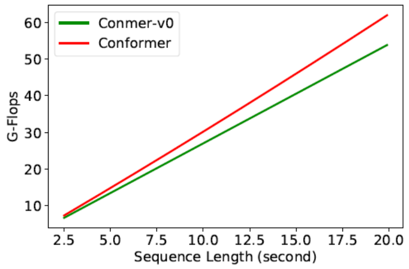

Figure 2: *The number of flops against the length of sequence (in seconds) measured for the Librispeech test-other data.*

3.3. The impact of MHSA on accuracy for short and long utterances

In order to investigate whether MHSA is more effective in recognizing longer utterances than the shorter ones, we conducted the following experiment. We chose the Librispeech test-other dataset which is considered to be the most challenging among three other evaluation datasets in this corpus. In this evaluation dataset, utterance length ranges from 2.5 to 20 seconds. We bucket the utterances into two groups, group1: $T \leq 10$ and group2: $T > 10$ seconds. We chose the length of 10 seconds because in most interactive voice assistants the length of utterance is around this range or shorter. We measure WER for each group for Conformer and Conmer-v0. We exclude samples for which WERs are equal or zero because we want to investigate where the two models make different predictions. We choose Conmer-v0 because we want to observe the impact of removing self-attention without adding any components to Conmer. We measured the WER for both causal (streaming) and non-causal (non-streaming) cases. Interestingly, as shown in Table 1, we found removing MHSA in the causal scenario reduces WER by 0.42% and 0.82 % for Conmer-v0 with smaller or comparable size of Conformer. For utterances whose length $>$ 10 sec, Conformer exhibits better performance suggesting MHSA is more effective for longer utterances. For the non-causal case,

Table 3: *WER (%) results on the Librispeech test data for the non-causal models.*

Model	dev-clean	dev-other	test-clean	test-other	size (M)
LSTM	4.71	12.73	4.81	13.07	30
Conformer	**3.46**	**8.75**	**3.72**	**8.81**	28
Conmer-v0	4.10	10.52	4.01	10.83	23
Conmer-v0 (d_{ff}=1,524)	3.80	10.60	4.00	10.72	29
Conmer-v1	**3.46**	9.72	**3.72**	9.75	28

Table 4: *Relative WER Reduction on de-identified in-house data compared to causal Conformer.*

Model	set 1	set 2	set 3	set4	set5	set 6	set 7	set 8	average
Conmer-v0	0.64	2.25	-0.96	-2.61	-2.98	1.93	4.49	-3.07	-0.08
Conmer-v1	2.29	-0.96	1.24	-2.24	-3.91	1.21	4.42	0.78	0.35
Conmer-v1 (Swish)	-0.19	1.67	1.41	-0.99	-1.63	1.59	9.83	1.20	1.95

we found the impact of MHSA is more pronounced and Conformer outperforms Conmer for both longer and shorter utterances. This is, however, expected as MHSA attends to both past and future frames to better recognize on the current one. These results suggest we can remove MHSA from the streaming Conformer transducer when dealing with short utterances without sacrificing accuracy. One reason as why Conmer works well for short utterances is that the depth-wise CNN layer in the CNN module takes the responsibility of MHSA and learns global context as wide as its kernel size (32 frames in our experiments) in absence of MHSA.

3.4. Conmer WER results

In order to obtain the overall performance of Conmer, we benchmark Conmer against Conformer and LSTM-based transducers. Tables 2 and 3 shows the WER results for all four sets of Librispeech evaluation datasets for both causal (streaming) and non-causal (non-streaming) cases. We observe for the causal case, Conformer outperforms Conmer narrowly by only 1% relative (9.32 vs. 9.42) when including all evaluation sets. On the non-causal scenario, Conformer, however, delivers significantly better accuracy (17.7% relative, 6.18, vs. 7.28), especially for test-other and dev-other. Again, this improvement is expected because this model benefits from a non-causal MHSA.

3.5. Computational complexity

We measured the number of floating point operations to compare the computational complexity of Conformer and Conmer. We used Electra package [31] and methods proposed in [32] to compute FLOPs for MHSA. For a CNN layer number of FLOPs is equal to $4c_ic_ok^2Td$, where the parameters, respectively, are the number of input and output channels, kernel size, sequence length and feature dimension. As shown in Figure 2, we found that Conmer-v0 has less G-FLOPs (on average 10% relative, 30.22 vs. 34.21) when tested on the Librispeech test-other dataset whose utterances's length are within 2.5 to 20 seconds. One G-FLOP is equal to one billion floating-point operations. We found two components play the major role in computational complexity: MHSA and FFM, respectively. In general, MHSA complexity is of order $O(d_{\text{attn}}T^2)$ where T and d_{attn} denote the length of sequence and the dimension of attention. FFM complexity is, however, linear with respect to the length and is of order of $O(d_{\text{model}}d_{ff})$, where d_{model} and d_{ff} denote the dimension of the model and hidden feedforward outputs. We found computational complexity of MHSA is comparable with FFM for short sequences for two reasons: first we downsample the sequence by order of four by the CNN frontend so the effective length of the sequence at the input of MHSA is $\frac{1}{4}n$. Second, $d_{ff} > d_{\text{attn}}$ in our setting. Consequently, we found Conmer-v1 has similar FLOPs compared to MHSA (average 34.41 vs 34.21) because each layer of Conmer-v1 deploys 3 FFMs compared to 2 for MHSA. Our results suggest adding FMM is not an effective way to increase accuracy because it reduces the efficiency.

Table 5: *Impact of quantization on WER for Librispeech data*

Model	dev-clean	dev-other	test-clean	test-other
Conformer	5.12	13.32	5.67	13.31
Conformer-8bit	5.54 (8.20%)	14.13	5.71	14.24
Conmer	5.28	13.37	5.60	13.52
Conmer-8bit	5.41	14.15	5.72	14.13
ConmerL-8bit	5.40	14.15	5.66	13.77

3.6. Conmer performance on real-world in-house data

In order to investigate how Conmer performs on real-world data, we also evaluate Conmer and its variant against Conformer on seven different sets of our de-identified in-house data. This data was collected from different traffics. Table 4 reports the relative WER reduction of each model when compared with Conformer. For in-house data, we observed using a Swish activation in the gating module further reduces WER. The relative numbers are calculated as $100 \times \frac{\text{WER}_{conformer} - \text{WER}_{conmer}}{\text{WER}_{conformer}}$. The results show that Conmer model performs on par or even better than Conformer for our in-house real-world data. These results further support our previous experiments conducted on the Librispeech dataset.

3.7. Robustness to quantization

To simulate the runtime environment, we enable quantization-aware-training [33] to both Conformer and Conmer, compressing the bit-width of all weights to 8-bit. Compared to Conmer-v0 trained in 32-bit floating point, Conmer-8bit achieves comparable performance on all dev and test datasets. It is also on par with Conformer-8bit with more parameters and attention modules, which indicates that Conmer is relatively robust to 8-bit model compression for hardware deployment. Furthermore, we add two extra Conmer layers in ConmerL to match the number of parameters of Conformer, yet with only marginal accuracy gain from Conmer-8bit.

4. Conclusion

In this paper, we introduced Conmer, a simplified version of Conformer with lower computational complexity and accuracy better than Conformer for interactive voice assistants. Conmer is a promising candidate for on-device interactive speech recognition where real-time responses are important and computational resources are limited. We show when length of utterances are short and causality is a requirement (for streaming applications), MHSA can be removed with no impact on accuracy but increase in efficiency. Finally, our computational complexity results suggest the use of FFM should be done with caution as this module introduces latency comparable to that of MHSA for short utterances. We also show Conmer is more quantization friendly which makes it a strong candidate for on-device applications where quantization is necessary to minimize footprint and execute on hardware accelerators.

5. References

[1] L. Dong, S. Xu, and B. Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *ICASSP 2018*. IEEE, 2018, pp. 5884–5888.

[2] N. Moritz, T. Hori, and J. Le, "Streaming automatic speech recognition with the transformer model," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6074–6078.

[3] 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 2020-2020*. IEEE, 2020, pp. 7829–7833.

[4] C.-F. Yeh, J. Mahadeokar, K. Kalgaonkar, Y. Wang, D. Le, M. Jain, K. Schubert, C. Fuegen, and M. L. Seltzer, "Transformer-transducer: End-to-end speech recognition with self-attention," *arXiv preprint arXiv:1910.12977*, 2019.

[5] A. Tripathi, J. Kim, Q. Zhang, H. Lu, and H. Sak, "Transformer transducer: One model unifying streaming and non-streaming speech recognition," *arXiv preprint arXiv:2010.03192*, 2020.

[6] X. Chen, Y. Wu, Z. Wang, S. Liu, and J. Li, "Developing real-time streaming transformer transducer for speech recognition on large-scale dataset," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5904–5908, 2020.

[7] W. Huang, W. Hu, Y. T. Yeung, and X. Chen, "Conv-transformer transducer: Low latency, low frame rate, streamable end-to-end speech recognition," *arXiv preprint arXiv:2008.05750*, 2020.

[8] F.-J. Chang, J. Liu, M. Radfar, A. Mouchtaris, M. Omologo, A. Rastrow, and S. Kunzmann, "Context-aware transformer transducer for speech recognition," *arXiv preprint arXiv:2111.03250*, 2021.

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

[10] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.

[11] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *Advances in neural information processing systems*, vol. 28, 2015.

[12] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang *et al.*, "Streaming end-to-end speech recognition for mobile devices," in *ICASSP*. IEEE, 2019, pp. 6381–6385.

[13] M. H. Radfar, R. Barnwal, R. V. Swaminathan, F.-J. Chang, G. P. Strimel, N. Susanj, and A. Mouchtaris, "Convrnn-t: Convolutional augmented recurrent neural network transducers for streaming speech recognition," *ArXiv*, vol. abs/2209.14868, 2022.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[15] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[16] W. Han, Z. Zhang, Y. Zhang, J. Yu, C.-C. Chiu, J. Qin, A. Gulati, R. Pang, and Y. Wu, "Contextnet: Improving convolutional neural networks for automatic speech recognition with global context," *arXiv preprint arXiv:2005.03191*, 2020.

[17] J. Li, V. Lavrukhin, B. Ginsburg, R. Leary, O. Kuchaiev, J. M. Cohen, H. Nguyen, and R. T. Gadde, "Jasper: An end-to-end convolutional neural acoustic model," *arXiv preprint arXiv:1904.03288*, 2019.

[18] S. Kriman, S. Beliaev, B. Ginsburg, J. Huang, O. Kuchaiev, V. Lavrukhin, R. Leary, J. Li, and Y. Zhang, "Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions," in *ICASSP*. IEEE, 2020, pp. 6124–6128.

[19] 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," *arXiv preprint arXiv:2005.08100*, 2020.

[20] S. Zhang, E. Loweimi, P. Bell, and S. Renals, "On the usefulness of self-attention for automatic speech recognition with transformers," *2021 IEEE Spoken Language Technology Workshop (SLT)*, pp. 89–96, 2020.

[21] J. Liu, Z. Pan, H. He, J. Cai, and B. Zhuang, "Ecoformer: Energy-saving attention with linear complexity," *ArXiv*, vol. abs/2209.09004, 2022.

[22] Z. Wu, Z. Liu, J. Lin, Y. Lin, and S. Han, "Lite transformer with long-short range attention," *ArXiv*, vol. abs/2004.11886, 2020.

[23] S. Lu, M. Wang, S. Liang, J. Lin, and Z. Wang, "Hardware accelerator for multi-head attention and position-wise feed-forward in the transformer," *2020 IEEE 33rd International System-on-Chip Conference (SOCC)*, pp. 84–89, 2020.

[24] H. Liu, Z. Dai, D. R. So, and Q. V. Le, "Pay attention to mlps," in *Neural Information Processing Systems*, 2021.

[25] J. Sakuma, T. Komatsu, and R. Scheibler, "Mlp-asr: Sequence-length agnostic all-mlp architectures for speech recognition," *ArXiv*, vol. abs/2202.08456, 2022.

[26] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[27] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[28] A. W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le, "Qanet: Combining local convolution with global self-attention for reading comprehension," *ArXiv*, vol. abs/1804.09541, 2018.

[29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*. IEEE, 2015, pp. 5206–5210.

[30] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[31] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," in *ICLR*, 2020. [Online]. Available: https://openreview.net/pdf?id=r1xMH1BtvB

[32] J. Kaplan, S. McCandlish, T. J. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," *ArXiv*, vol. abs/2001.08361, 2020.

[33] K. Zhen, M. H. Radfar, H. D. Nguyen, G. P. Strimel, N. Susanj, and A. Mouchtaris, "Sub-8-bit quantization for on-device speech recognition: A regularization-free approach," *2022 IEEE Spoken Language Technology Workshop (SLT)*, pp. 15–22, 2022.