Do we really need Self-Attention for Streaming Automatic Speech Recognition?

Youness Dkhissi, Valentin Vielzeuf, Elys Allesiardo, Anthony Larcher

DO WE REALLY NEED SELF-ATTENTION FOR STREAMING AUTOMATIC SPEECH RECOGNITION?

Youness Dkhissi[1,2]*, Valentin Vielzeuf*[1]*, Elys Allesiardo*[1]*, Anthony Larcher*[2]

1. Orange Innovation {firstname}.{lastname}@orange.com
2. LIUM, Le Mans Université Avenue Olivier Messiaen,72085 Le Mans CEDEX 9, France

ABSTRACT

Transformer-based architectures are the most used architectures in many deep learning fields like Natural Language Processing, Computer Vision or Speech processing. It may encourage the direct use of Transformers in the constrained tasks, without questioning whether it will yield the same benefits as in standard tasks.

Given specific constraints, it is essential to evaluate the relevance of transformer models. This work questions the suitability of transformers for specific domains. We argue that the high computational requirements and latency issues associated with these models do not align well with streaming applications. Our study promotes the search for alternative strategies to improve efficiency without sacrificing performance.

In light of this observation, our paper critically examines the usefulness of transformer architecture in such constrained environments. As a first attempt, we show that the computational cost for Streaming Automatic Speech Recognition (ASR) can be reduced using deformable convolution instead of Self-Attention. Furthermore, we show that Self-Attention mechanisms can be entirely removed and not replaced, without observing significant degradation in the Word Error Rate.

Index Terms— streaming automatic speech recognition, self-attention, conformer, deformable convolution

1. INTRODUCTION

Transformers[1] have become the *de facto* architecture for Natural Language Processing (NLP). Most NLP solutions build on top of a model pre-trained on a large dataset to take advantage of Self-Attention mechanisms and capture the long-range dependencies between the input sequence elements. After this initial pre-training, the model can be fine-tuned on the downstream tasks. The gain of performance brought by Transformers has encouraged the use of this architecture and its adaptation for other fields.

Vision Transformer[2] (ViT) have ended the dominance of Convolutional Neural Network (CNN) based architectures in the State Of The Art (SOTA) of many Computer Vision tasks. Also, in the Speech field, the integration of the Self-Attention mechanism has been very effective in improving the SOTA on different tasks. In particular, for Automatic Speech Recognition (ASR), Conformers[3] have become the SOTA architecture thanks to the combination of Self-Attention modules that capture long-range dependencies and Convolutional modules that capture short-range dependencies.

This great success of the Self-Attention-based architectures encourages the community to directly re-use them in more constrained applications of these tasks. Streaming Automatic Speech Recognition[4] is an example of such a task, where models should begin to transcribe the speech input into text without having the full speech context. This means that models in Streaming ASR process speech input as chunks and do not take the full-context of speech while transcribing. Using small chunks while training the ASR models limits the visibility of the Self-Attention within these chunks. The result is a potential mismatch between architecture and task: the Self-Attention module, designed to exploit global context, is forced to operate locally while keeping its heavier cost profile.

For this specific task, many recent works bring improvements in terms of transcription quality and latency of the models[5, 6, 7, 8, 9], but the proposed improvements never question architectures that have been designed for standard task, i.e., non-streaming ASR. [10] and [11] show that replacing Self-Attention by a linear alternative that summarizes input information into a global vector or a Mamba module[12] in Streaming ASR architectures preserves the performance of the system. However, the reason why Self-Attention does not bring much improvement in Streaming ASR and why it could be replaced by alternative modules remains unexplored by the community, to the best of our knowledge.

This paper addresses these questions from the standpoint of encoder design for streaming ASR. We adopt a strict streaming setup in which models are trained and evaluated with fixed chunk sizes and no access to past context beyond the current chunk. Within this regime, we first analyse how Self-Attention actually behaves inside a Conformer encoder. Visualizing layer-wise mean attention maps with chunked inference reveals dominant near-diagonal patterns, indicating that attention is predominantly capturing short-range dependencies already well handled by the convolution module. This finding suggests that, under streaming constraints, self-attention effectively operates as an expensive local operator.

Guided by this analysis, we revisit the encoder architecture and study two pragmatic modifications that reduce cost while preserving accuracy. In a *soft* approach, we substitute Self-Attention with a lightweight 1-D deformable convolution module, which adaptively focuses on local patterns within each chunk while maintaining the conformer's overall structure. In a *hard* approach, we remove Self-Attention, relying on the Conformer's convolution module to capture local and chunk-level patterns. These designs are evaluated in a Conformer-Transducer framework on LibriSpeech and TEDLIUM-2 across multiple chunk sizes (160–1280 ms), using strictly streaming training/inference.

2. PRELIMINARY ANALYSIS

In this Section, we perform a preliminary analysis by examining how Self-Attention behaves in a strict streaming setting within a Conformer-Transducer architecture, which we consider as a baseline in the whole paper. We choose the transducer architecture[13] as it appears to be more suitable for streaming ASR as shown in

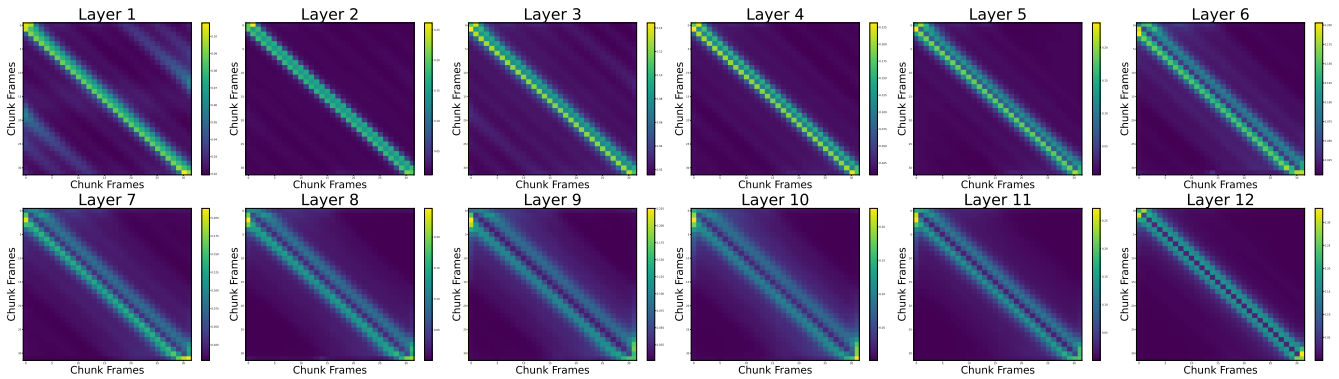


Fig. 1: Illustration of the mean attention map per layer using chunk size 1280 ms on LibriSpeech test-clean dataset.

Table 1: Word Error Rate (WER) measurements on a chunk size of 1280 ms with attention masking. The values in brackets correspond to the absolute difference between the result and the baseline with no masking.

Number of non-masked diagonals	LibriSpeech test-clean	LibriSpeech test-other
all	3.36	8.91
7	3.84(+0.48)	9.96(+1.05)
5	4.17(+0.81)	10.81(+1.90)

[14, 15, 16]. We adopt a strict chunked regime at both training and inference: the model processes fixed-size chunks with no access to past or future context beyond the current chunk.

Figure 1 shows the mean attention map calculated for each layer of a 12-layer conformer encoder in our baseline. We train and test this model using a chunk size of 1280 ms (32 audio frames). We see that, across all layers, attention concentrates in narrow bands around the main diagonal, with the strongest weights near the centre of the chunk. This pattern indicates that, in the streaming mode, Self-Attention predominantly captures short-range dependencies within each chunk. Given that the Conformer's convolution module uses a kernel size of 31 and is stacked across layers, its effective receptive field within a 32-frame chunk can span most of the chunk. As a result, the convolution module likely aggregates chunk-level information, effectively reversing the canonical roles described for full-context Conformers, where Self-Attention models global dependencies and convolution focuses on local patterns.

To empirically confirm this hypothesis, we test, **without fine-tuning**, the baseline while masking all attention maps except their central diagonals. We obtain this configuration by masking the attention maps, before applying Softmax function, using the following mask M where N_{diag} is the number of the non-masked central diagonals.

$$M_{i,j} = \begin{cases} 1, & \text{if } |i-j| \leq \left\lfloor \frac{N_{diag}}{2} \right\rfloor \\ 0, & \text{otherwise} \end{cases}$$

Table 1 shows the results using this configuration while keeping only 7 or 5 non-masked diagonals in the attention maps, i.e., keeping, respectively, 21% or 15% of the values of each attention map. Despite the expected degradation, the model sustains reasonable performance, supporting the interpretation that, under strict streaming constraints, Self-Attention module operates primarily as an expensive local operator, while the convolution module carries much of the chunk-level information.

These observations suggest two pragmatic directions: replacing self-attention with a lighter local operator tuned for chunked inputs, or removing it altogether and relying on the convolution module to capture both local and chunk-level patterns. We explore both options in the next section.

3. ATTENTION REPLACEMENT

Motivated by the preliminary analysis, we revisit the encoder design, in this Section, to show how much Self-Attention contributes under strict chunked streaming. We use the conformer encoder as a baseline of the model used in Streaming ASR task. This encoder is composed of a convolution subsampling block and a stack of conformer blocks. Each conformer block features a sandwich architecture, with a feed-forward module positioned at both the beginning and the end, enclosing a Self-Attention module and a convolutional module within. In our experiments, we explore two approaches to replace self-attention and evaluate its effectiveness.

Note that we deliberately do not adopt efficient attention variants such as Fastconformer[17], Linformer-style linear attention[18], local/blockwise attention, or memory-based streaming Transformers (Emformer[19], Zipformer[20]) for two reasons. First, our goal is to isolate the intrinsic contribution of self-attention under strict streaming constraints, where no past context or look-ahead is available. Many of these approaches explicitly rely on external memory/caches. Including them would obscure whether attention itself remains useful in this regime. Second, even efficient attention variants have non-trivial constant factors and implementation complexity that differ from standard attention and from convolutions; comparing them fairly would require substantial engineering and careful latency-quality trade-off tuning beyond the scope of this first study. Our objective here is to establish a clear, controlled baseline result: in strict chunked streaming, attention behaves primarily as a local operator and can be replaced by lightweight convolutional modules, or even removed, without significant loss in accuracy.

3.1. Hard Approach

In this approach, we completely remove the Self-Attention module without modifying the rest of the architecture. We suppose that the convolution module is solely capable of extracting all the diagonal patterns that could be captured by the Self-Attention inside the chunk due to the fact that its kernel size, 31 as in [3], is larger than or equal to the chunk sizes tested in streaming inference.

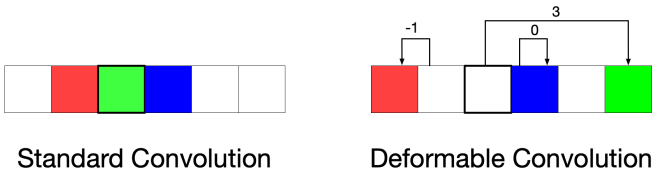


Fig. 2: Illustration of a 1-D standard convolution (left) and 1-D deformable convolution (right) with kernel size of 3 applied at the 3^{rd} timestep. The coloured squares represent the elements on which the kernel of each convolution will be applied. The offsets applied to the deformable convolution in this example are [-1,3,0].

3.2. Soft Approach

This approach consists of replacing the Self-Attention module with a deformable convolution module composed of a 1-D deformable convolution[21] followed by a layer normalisation and a Swish activation layer.

The deformable convolution, which is the core of our module, is a type of convolution with an asymmetric kernel. It works in 2 steps: first an offset convolution is applied on the input to predict input position offsets. Then, an output convolution is applied to the input taking into account the predicted input position offsets to shift its kernel elements, as shown in Figure 2.

We choose using a deformable convolution rather than a standard convolution because the standard convolution processes all regions equally when a convolution kernel slides across the input audio where a deformable convolution can help build local patterns by focusing on the most relevant information.

4. EXPERIMENTS

4.1. Evaluation protocol

We conduct our experiments on two widely used public datasets: LibriSpeech[22] that contains 960 hours of read English speech and TEDLIUM-2 [23] that contains 207 hours of TED Talks in order to show the effectiveness of our approach on more spontaneous speech.

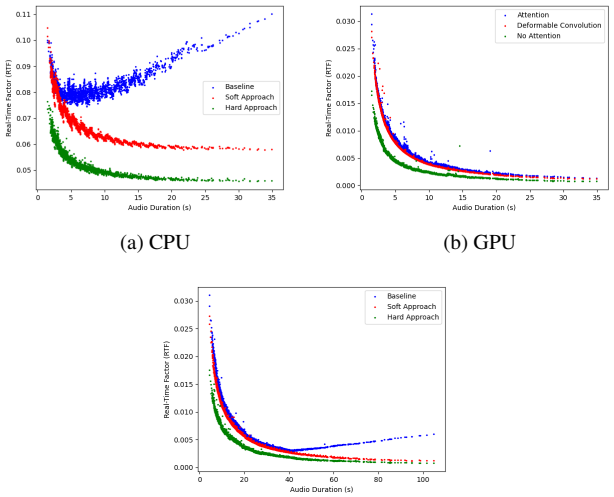
To evaluate our approach on the Streaming-ASR task, we use the Word Error Rate metric. Also, to show the significance of the results, each result of our baseline is presented together with its confidence interval[1]. These intervals are calculated using the bootstrapping method with 1,000 bootstrap sets. They were also calculated between 2.5 and 97.5 percentiles to exclude outliers.

4.2. Model configuration

All experiments were conducted using the SpeechBrain toolkit[24]. The baseline used for Streaming-ASR task and the proposed systems is Conformer Transducer which share the following components: **(a) 2 Convolutional layers** with kernel size of 2 and stride of 2, which downsamples the frame rate by 4. **(b) 12-layer Conformer encoder**[3] having 512-dimensional input where each layer is composed of: feed-forward network of size 2048, convolution block having kernel size of 31 with stride of 1 and a self-attention block with 8 attention heads in the case of the baseline. In the soft approach, we replace it with 1-D deformable convolution that has a kernel size of 5 and uses 8 groups.[2] **(c) Predictor network** of 1-layer LSTM[25] with hidden size of 512.

[1]https://github.com/luferrer/ConfidenceIntervals

[2]https://github.com/inspiros/tvdcn/tree/master/tvdcn



(a) CPU (b) GPU

(c) GPU on augmented dataset

Fig. 3: Comparison of Real Time Factor (RTF) between the baseline, soft and hard approach for each utterance on LibriSpeech test-clean dataset. In 3c, the utterances have been repeated to be x3 longer.

During the training stage, we train our models for 150 epochs using transducer loss and an auxiliary CTC loss[26] for the first 10 epochs. In addition, an Adam optimizer is set with a learning rate of 0.0008 and a weight decay of 0.01. All our models are trained and tested using a unique chunk size without taking into account any past context.

4.3. Results

Table 2 shows the Word Error Rate obtained by the different approaches compared to the baseline in different chunk sizes. We first see that using soft or hard approach implies an important reduction in terms of the number of parameters. Moreover, both approaches present insignificant degradation (in the sense of confidence intervals) compared to the baseline. The soft approach even outperforms the baseline while testing with small chunk sizes. This may be explained by the great capacity of deformable convolutions to capture local patterns compared to the Self-Attention module.

To compare the efficiency of the proposed approaches in terms of computational cost, we measure the Real Time Factor (RTF) for each utterance of the LibriSpeech test-clean dataset, as we illustrate in Figure 3. We use the same method as in [10], by calculating the RTF on the encoder only, because the other components stay the same for all trained models. The measurements were made using the CPU *AMD EPYC 7282-2.8GHz* and the GPU *RTX 4070*. We observe that both soft and hard approaches reduce the computational cost of processing the utterances compared to the baseline. In fact, while using CPU (Figure 3a), the mean RTF across the dataset for the baseline is $8.18.10^{-2}$ where the soft approach gives an RTF of $6.89.10^{-2}$, which is 15.8% faster. For the hard approach, it has an RTF of $4.51.10^{-2}$ which makes it 44.9% faster than the baseline. For the GPU, the mean RTF across the dataset for the baseline is $8.21.10^{-3}$ where the soft approach gives an RTF of $7.78.10^{-3}$, which is 5.2% faster. For the hard approach, it has an RTF of $4.51.10^{-3}$ which makes it approximately 2 times faster than the baseline.

On GPU, Figure 3b, the quadratic scaling of self-attention with

Table 2: the Word Error Rate measurements on the LibriSpeech test-clean, test-other and TEDLIUM-2 datasets with different chunk sizes. The values in brackets correspond to the relative reduction compared to the baseline and the values in square brackets represent the confidence interval of the result.

Dataset	Model	Number of parameters	Chunk size			
			160ms	320ms	640ms	1,280ms
LibriSpeech test-clean	Baseline	81.3M	4.21 [3.98;4.47]	3.85 [3.62;4.10]	3.69 [3.47;3.92]	3.36 [3.15;3.58]
	Soft approach	67.6M(**-16.8%**)	**4.11** [3.90;4.35]	3.86 [3.64;4.09]	3.75 [3.54;3.98]	3.56 [3.35;3.80]
	Hard approach	65.5M(**-19.4%**)	4.29 [4.06;4.53]	4.04 [3.81;4.29]	3.78 [3.56;4.00]	3.62 [3.41;3.84]
LibriSpeech test-other	Baseline	81.3M	11.06 [10.59;11.49]	10.36 [9.92;10.79]	9.78 [9.38;10.18]	8.91 [8.52;9.28]
	Soft approach	67.6M(**-16.8%**)	**11.03** [10.62;11.43]	**10.33** [9.90;10.74]	9.81 [9.37;10.23]	9.34 [8.94;9.72]
	Hard approach	65.5M(**-19.4%**)	11.23 [10.78;11.64]	10.39 [9.98;10.79]	9.81 [9.42;10.21]	9.62 [9.24;10.00]
TEDLIUM-2	Baseline	81.3M	11.12 [10.56;11.66]	10.32 [9.82;10.80]	9.77 [9.28;10.23]	9.26 [8.79;9.73]
	Soft approach	67.6M(**-16.8%**)	**11.04** [10.52;11.60]	10.49 [9.98;10.99]	10.06 [9.55;10.55]	9.27 [8.81;9.73]
	Hard approach	65.5M(**-19.4%**)	11.47 [10.93;11.99]	11.08 [10.54;11.59]	10.29 [9.81;10.81]	9.87 [9.35;10.34]

Table 3: Ablation study on the number of parameters using chunk sizes 320 ms and 640 ms. Performance are given in terms of WER and the values in brackets correspond to the absolute difference compared to the baseline in Table 2.

Model	Number of parameters	Chunk size	LibriSpeech test-clean	LibriSpeech test-other
Soft approach	79.7M	320 ms	3.77(**-0.08**)	10.22(**-0.14**)
		640 ms	3.73(+0.04)	9.62(**-0.16**)
		1280 ms	3.44(+0.08)	9.15(+0.24)
Hard approach	79.6M	320 ms	3.92(+0.07)	10.44(+0.08)
		640 ms	3.60(**-0.09**)	9.84(+0.06)
		1280 ms	3.52(+0.16)	9.18(+0.27)

sequence length is largely masked by kernel-level parallelism and highly optimized Transformer implementations. To expose this behaviour, we extended the input by repeating each test utterance three times, as shown in Figure 3c. Under this long-utterance regime, the Self-Attention cost becomes evident, like in CPU, with the quadratic trend emerging for inputs longer than 45 seconds. Consequently, for datasets with long recordings, the soft and hard approaches deliver larger RTF reductions than the baseline on GPU.

4.4. Ablation studies

As shown in Table 3, we conducted the first ablation studies to show the impact of the architecture compared to the number of parameters used in the model. For that, we train the soft and hard approaches using a number of parameters equivalent to the baseline. We achieve this by increasing the embedding dimension. We observe from Table 3 that the soft approach surpasses the baseline even when using larger chunk sizes. Moreover, the hard approach matches the baseline results for these large chunk sizes, which confirms that the convolution module alone is capable of capturing the local and global patterns inside the chunk. In conclusion, the insignificant degradations observed in Table 2 in larger chunk sizes are not related to the Self-Attention module but to the number of parameters.

Table 4: Ablation study on deformable convolution kernel sizes, used in the soft approach, trained on 1280 ms chunk size.

Model	Kernel size	LibriSpeech test-clean	LibriSpeech test-other
Soft approach	5	3.56	9.34
	9	3.54(-0.02)	9.49(+0.15)
	17	3.51(-0.05)	9.04(-0.30)

We conducted a second ablation study, described in Table 4, to analyse the effect of the chosen kernel size for the deformable convolution module. We observe that augmenting the kernel size does not bring a significant improvement in LibriSpeech test-clean dataset and inconsistent results when testing on LibriSpeech test-other. These results confirm again that the deformable convolution needs to capture, in most cases, the short-range dependencies, and it does not require a wide kernel.

5. CONCLUSION

Our study reveals that Self-Attention mechanisms in streaming ASR models may not be as essential as previously assumed, and found that it behaves as a local operator: attention maps concentrate near the diagonal, and constraining them to narrow central bands yields only minor WER changes. Guided by these findings, we proposed two encoder variants within a Conformer-Transducer, soft and hard approaches, that garde the performances or even surpass Attention-based model with significant computational cost and parameters reduction. Future work should focus on developing architectures specifically designed for streaming ASR that prioritise local information processing. Given our findings, architectures that replace Self-Attention with more efficient modules like specialised convolutional modules, inspired by architectures such as Contextnet[27] and Quartznet[28], should be explored.

6. ACKNOWLEDGEMENTS

This work was also granted access to the HPC resources of IDRIS under the allocation 2025-A0191014876 made by GENCI.

7. REFERENCES

[1] A Vaswani, "Attention is all you need," *Neurips*, 2017.

[2] Alexey Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[3] Anmol Gulati et al., "Conformer: Convolution-augmented transformer for speech recognition," in *Interspeech 2020*, 2020, pp. 5036–5040.

[4] Ehsan Variani et al., "Global normalization for streaming speech recognition in a modular framework," *Neurips*, vol. 35, pp. 4257–4269, 2022.

[5] Jiahui Yu et al., "Dual-mode asr: Unify and improve streaming asr with full-context modeling," in *International Conference on Learning Representations*, 2021.

[6] Jiahui et al. Yu, "Fastemit: Low-latency streaming asr with sequence-level emission regularization," in *ICASSP 2021-2021*. IEEE, 2021, pp. 6004–6008.

[7] Wei Kang et al., "Delay-penalized transducer for low-latency streaming asr," in *ICASSP 2023-2023*. IEEE, 2023, pp. 1–5.

[8] Xingchen Song et al., "Trimtail: Low-latency streaming asr with simple but effective spectrogram-level length penalty," in *ICASSP 2023-2023*. IEEE, 2023, pp. 1–5.

[9] Emiru Tsunoo et al., "Decoder-only architecture for streaming end-to-end speech recognition," in *Interspeech 2024*, 2024, pp. 4463–4467.

[10] Titouan Parcollet et al., "Linear time complexity conformers with summarymixing for streaming speech recognition," in *ICASSP 2025 - 2025*, 2025, pp. 1–5.

[11] Takafumi Moriya et al., "Attention-free dual-mode asr with latency-controlled selective state spaces," in *Proc. Interspeech 2025*, 2025, pp. 3588–3592.

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

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

[14] Shashi Kumar et al., "Xlsr-transducer: Streaming asr for self-supervised pretrained models," in *ICASSP*. IEEE, 2025.

[15] Bo Li et al., "A better and faster end-to-end model for streaming asr," in *ICASSP*. IEEE, 2021.

[16] Bo Li, Shuo-yiin Chang, Tara N Sainath, Ruoming Pang, Yanzhang He, Trevor Strohman, and Yonghui Wu, "Towards fast and accurate streaming end-to-end asr," in *ICASSP*, 2020.

[17] Dima Rekesh et al., "Fast conformer with linearly scalable attention for efficient speech recognition," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.

[18] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma, "Linformer: Self-attention with linear complexity," *arXiv preprint arXiv:2006.04768*, 2020.

[19] Yangyang Shi, Yongqiang Wang, Chunyang Wu, Ching-Feng Yeh, Julian Chan, Frank Zhang, Duc Le, and Mike Seltzer, "Emformer: Efficient memory transformer based acoustic model for low latency streaming speech recognition," in *ICASSP*. IEEE, 2021.

[20] Zengwei Yao, Liyong Guo, Xiaoyu Yang, Wei Kang, Fangjun Kuang, Yifan Yang, Zengrui Jin, Long Lin, and Daniel Povey, "Zipformer: A faster and better encoder for automatic speech recognition," *CoRR*, 2023.

[21] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei, "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 764–773.

[22] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015*. IEEE, 2015, pp. 5206–5210.

[23] Wei Zhou, Wilfried Michel, Kazuki Irie, Markus Kitza, Ralf Schlüter, and Hermann Ney, "The rwth asr system for ted-lium release 2: Improving hybrid hmm with specaugment," in *ICASSP*. IEEE, 2020.

[24] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, et al., "Speechbrain: A general-purpose speech toolkit," *arXiv preprint arXiv:2106.04624*, 2021.

[25] Alex Graves and Alex Graves, "Long short-term memory," *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.

[26] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006.

[27] Wei Han, Zhengdong Zhang, Yu Zhang, Jiahui Yu, Chung-Cheng Chiu, James Qin, Anmol Gulati, Ruoming Pang, and Yonghui Wu, "Contextnet: Improving convolutional neural networks for automatic speech recognition with global context," 2020.

[28] Samuel Kriman, Stanislav Beliaev, Boris Ginsburg, Jocelyn Huang, Oleksii Kuchaiev, Vitaly Lavrukhin, Ryan Leary, Jason Li, and Yang Zhang, "Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions," in *ICASSP*. IEEE, 2020.