INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

miniStreamer: Enhancing Small Conformer with Chunked-Context Masking for Streaming ASR Applications on the Edge

Haris Gulzar[1], *Monikka Roslianna Busto* [1], *Takeharu Eda*[1], *Katsutoshi Itoyama*[2], *Kazuhiro Nakadai*[2]

[1]NTT, Software Innovation Center, Japan, [2]Tokyo Institute of Technology, Japan

(haris.gulzar.cf, monikkaroslianna.busto.px, takeharu.eda.bx)@hco.ntt.co.jp,
(itoyama, nakadai)@ra.sc.e.titech.ac.jp

Abstract

Real-time applications of Automatic Speech Recognition (ASR) on user devices on the edge require streaming processing. Conformer model has achieved state-of-the-art performance in ASR for the non-streaming task. Conventional approaches have tried to achieve streaming ASR with Conformer using causal operations, but it leads to quadratic increase in the computational cost as the utterance length increases. In this work, we propose a chunked-context masking approach to perform streaming ASR with Conformer, which limits the computational cost from quadratic to a constant value. Our approach allows self-attention in Conformer encoder to attend the limited past information in form of chunked context. It achieves close to the full context causal performance for Conformer-Transducer, while significantly reducing the computational cost and maintains a low Real Time Factor (RTF) which is highly desirable trait for resource-constrained low-power edge devices.

Index Terms: streaming ASR, attention masking, ASR for resource-constrained devices

1. Introduction

Speech is one of the most preferred interfaces to interact with intelligent devices [1]. As the applications of AI on user devices continue to grow [2, 3], so does the demand to deploy computationally efficient Automatic Speech Recognition (ASR) models on low-power edge devices. In several applications, real-time speech recognition is desired, meaning the text is predicated on the go as the speaker utters speech. This mode of speech is generally referred to as streaming or online ASR [4]. Until recently, Recurrent Neural Network (RNN) based models were the de-facto choice for streaming ASR, because of their ability to keep the context in the RNN state from previous inputs [5, 6, 7]. The introduction of the Transformer in ASR [8] has shifted the focus from RNN to training-friendly Transformer encoder-decoder-based models.

Many state-of-the-art (SOTA) end-to-end ASR models based on Transformer are proposed [9, 10] to perform speech recognition in a sequence-to-sequence manner, meaning that they predict the full-text sequence by consuming a complete speech signal. These models require the speaker to finish speaking before they can start predicting the text. Convolution augmented Transformer (Conformer) encoder is proposed [10] which synergizes the local and global features extraction by convolution and attention layers respectively, and achieved SOTA performance for non-streaming ASR. Conformer is proposed in three different sizes and the Conformer-Small is best suitable for ASR on resource-constrained devices.

Recently, a few techniques are proposed to enable Transformer based models like Conformer to work for streaming applications. One way is to make the layers of the encoder causal [11], meaning the attention layers in the encoder are allowed to attend only to past and current frames in the sequence. This future-restricted attention approach has to process the entire past sequence repeatedly as the new frames arrive. Therefore the computational cost of such a streaming set-up continues to grow as the length of utterance increases, given the quadratic computational complexity of the self-attention layer [12]. Another approach is to perform ASR in a chunk-based setting [13], which processes a chunk of audio frames through the encoder and then performs decoding on the encoded chunk. Chunked based streaming approach results in significant performance degradation as the attention layer is only allowed to attend the elements inside the given chunk.

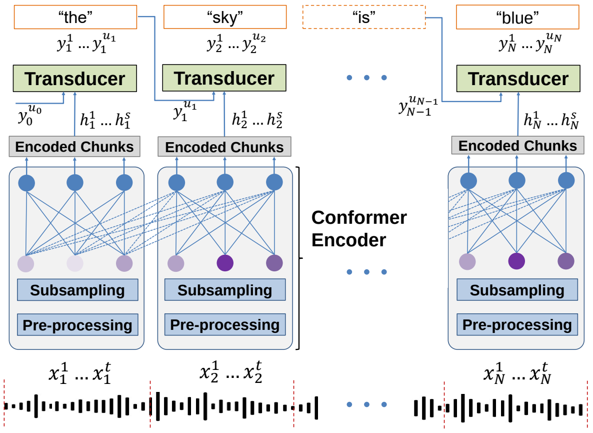


Figure 1: *Overview of our proposed streaming Conformer-Transducer architecture. Conformer consumers chunks of audio where attention layer attends the frames from current and past chunks. The Transducer decodes the latest encoded chunk while using the last predicted label from previous chunk.*

Our work proposes a context based chunk masking, which bridges the gap between full context causal attention and chunk-based attention approaches for streaming ASR. Chunked-context mask allows attention layer in Conformer to attend not only elements in the current chunk but also from a predefined number of previous chunks. Overall our work makes the following contributions:

- It proposes a chunk-based context masking technique to enable Conformer-Small to work for streaming ASR with limited computational cost and low latency.
- It demonstrates the real-time deployment of streaming ASR with Conformer encoder and Transducer decoder.

10.21437/Interspeech.2023-1162

2. Related Work

This section discusses related work in two aspects. First is the general work related to the deployment of ASR models on edge devices, and the second is about the proposed methods for enabling streaming ASR with Transformer-based models.

For ASR applications on the edge, mainly RNN-based encoder-decoder models are proposed [14, 15, 16, 17]. RNN-Transducer [14] model was proposed recently with a reasonably small size of 35M parameters for edge applications. [15] discussed the detailed performance analysis of several RNN models on edge devices. [16] uses knowledge distillation to obtain a streaming model from a non-streaming RNN-Transducer. A model optimization technique in [17] is used for RNN-Transducer to optimize the model size for edge devices. However, the challenge with RNN-based models is that their performance degrades for long utterances [18].

Transformer-based models especially Conformer have significantly outperformed RNN-based models for end-to-end ASR tasks [19]. Some recent proposals have enabled Conformer to work for streaming ASR. A recent work [20] based on causal masking of the wav2vec2.0 model is proposed, which works by updating the state of the encoder for every new chunk of speech frames. Another work [21] achieves streaming ASR of Conformer with future restricted self-attention. A combined triggered attention-based CTC and RNN decoder mechanism is proposed in [22] which further improves the streaming ASR performance with a causal Transformer encoder. However, the causal mechanism of the encoder causes a quadratic increase in the computational cost as the length of utterance increases. A sequentially sampled chunk-based streaming approach is proposed in [23] for Conformer encoder with Connectionist Temporal Classification (CTC) [24] and Transformer decoders. However, the computational cost of the encoder still increases linearly for long utterances. Secondly, CTC and Transformer decoders are not suitable for single-chunk decoding as they fail to make a connection between consecutive chunks.

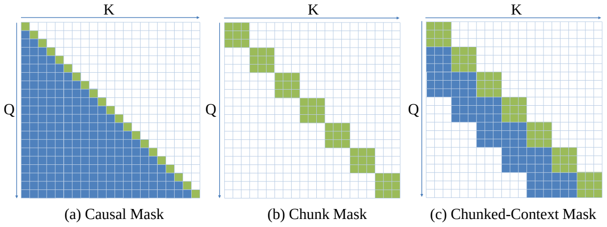
Our proposed Conformer encoder performs streaming ASR at a fixed computational cost and uses the Transducer decoder to perform streaming decoding only for the latest chunk while keeping a connection with the previous chunks.

3. Proposed Architecture

3.1. Chunked-Context Masking

The sequence-to-sequence modeling ability of the Conformer model comes from the inherent nature of the self-attention layer. Each element of the sequence (query) computes its attention score with respect to all other elements of the sequence (keys). For the speech recognition task, self-attention requires a full sentence to be spoken to compute attention scores, hence restricting transformers to be applied for streaming ASR without any modification. The masking technique can restrict the self-attention layer to attend only specific frames of the sequence during the training. At inference time, it can process the incoming chunk of frames from speech sequence of the same length as it was trained.

A causal mask for self-attention as in Figure 2 (a), allows each new frame or query (Q) in sequence to attend all past frames or keys (K) as proposed by [20, 21]. Figure 2 (b) shows a chunk mask which allows only a chunk of audio to compute attention scores with one another as in [13]. Our proposed masking approach named Chunked-Context Masking allows replicating a chunked-context inference set-up in the training by shifting the context mask window for the current and past chunks as shown in Figure 2 (c). Our mask allows queries to attend keys from the current chunk and also predefined number of past chunks for computing attention scores for each frame. ASR is basically a monotonic alignment task [25], meaning that, elements in the sequence that are very far from the current frame are less relevant. To make use of this inherent monotonic nature, discarding past frames beyond a certain context is a win-win scenario for ASR on resource-constrained edge devices.



(a) Causal Mask (b) Chunk Mask (c) Chunked-Context Mask

Figure 2: *Different types of masks for streaming Conformer. Green shows new frames for the current chunk, blue shows the attended context, and white shows unattended frames. (a) is a causal mask, when applied in a streaming setup, it computes the attention for the entire past sequence as the new frame arrives. (b) is the chunk-based approach which performs attention only for frames in the current chunk. (c) is our proposed approach that computes attention while attending to context chunks and the current chunk.*

Supposing the length of the input sequence is X, and the dimension of the encoder is D. The complexity of Multi-Head Self Attention (MHSA) is linear to dimension D and quadratic to the length of sequence X. On top of that, before computing the self-attention, linear operations for Query (Q), Key (K), Value (V), and Output (O) need to be computed and they have linear complexity with respect to X and quadratic to D. Overall computational cost of MHSA module, represented by Ω can be written as Equation 1 .

$$\Omega(\text{MHSA}) = 4XD^2 + X^2D \tag{1}$$

On the other hand, if we divide input sequence of length X into chunks of size C each, the complexity of MHSA will be quadratic to chunk size C multiplied with the number of chunks, n. Overall the complexity of our Chunked Context Mask-based MHSA (CCM_MHSA) module will be as represented by Equation 2. For a fixed number of chunks n and size of the chunk C, the computational cost would remain constant irrespective of the length of sequence as also shown in Figure 3.

$$\Omega(\text{CCM\_MHSA}) = 4CD^2 + (nC)^2D \tag{2}$$

3.2. Chunk Encoder with Transducer (miniStreamer):

After training the model with proposed chunk mask, it processes the inference on audio stream as shown in Figure 1. The chunk of audio of having t frames in each chunk are processed through the encoder. If we represent the N_{th} input chunk vector having C elements as x_N, the encoder will take n number of chunks ranging from x_{N-n} to x_N as input and will produce the corresponding encodings as h_{N-n} to h_N.

$$h_{N-n}...h_N = encoder(x_{N-n}...x_N) \tag{3}$$

Where n is number of chunks including current chunk and context chunks. In his way, the maximum number of audio frames for encoder would be $n * t$ irrespective of the total length of audio. The encoded chunks for the latest chunk h_N will then be fed to a decoder to predict text labels.

Several decoding approaches are proposed with Conformer encoder including CTC [24] Transformer [26] and Transducer [27]. In the case of chunk-based streaming ASR, the network requires to perform decoding only for the newest encoded elements from the current chunk. As for the CTC and Transformer decoders, they perform decoding on the chunks independently with no information about the previous frames, hence using these decoders will not result in appropriate predictions at the edges of each chunk.

We use a Transducer decoder which consists of a small 2-layer RNN as a predictor network and a single fully connected layer-based joint network. Transducer decoding keeps a connection between the last predicted label from the previous chunk and the following tokens in the next chunk. Given that Transducer produces u_N number of tokens for N_{th} chunk. This decoding step will be represented by Equation 4.

$$y_N = transducer(y_{N-1}^{u_{N-1}}, h_N) \qquad (4)$$

This step is shows in Figure 1 as green block. It is worth mentioning that, streaming decoding approaches based CTC or Transformer based decoders, perform decoding on entire encoded sequence at each step, which adds a significant computational overhead. Our proposed architecture performs decoding only on the encodings of latest chunk, irrespective of context length in encoder which results in fastest decoding time. It is also important to reiterate the significance of change in dimensions at different stages of the network. The signal x_N in each chunk of audio signal has t number of frames, which are then pre-processed after which one frame represents a 10ms audio signal. These frames are down-sampled by 8 times by following sub-sampling modules after which each signal represents a 80ms signal. The encoder embeddings have fixed dimension of s for each chunk. However, the decoder may produce a variable number of text labels to align with each audio chunk, hence represented by u_N at the final stage. The parametric details of each stage of the network are explained in the following section.

4. Experimental Results

4.1. Network Details:

We used the small version of the Conformer[10] as encoder in our experiments. The first stage of the network is audio pre-processing, to convert a uni-dimensional temporal audio signal into 80 MFCC features. We used a 10 ms hop length and a window length of 25 ms. We also apply spectrogram augmentation [28] with two masks in each frequency and time dimensions. Next, we have three 2D convolution sub-sampling layers with kernel size 3, and 144 filters in each layer. We used layer normalization in each sub-sampling module and this stage of the network down-samples the audio from 10 ms to 80 ms per element of the sequence, which becomes the input of the MHSA layer of the following 16 consecutive Conformer blocks. Each Conformer block has a convolution module and MHSA layer sandwiched between feed-forward modules as proposed in the original Conformer [10]. We also used sinusoidal positional encoding relative to the start of each utterance, before feeding input elements to Conformer blocks. For the Transducer decoder, we used 2 layers of RNN as a sub-word predictor network and a fully connected layer-based joint network to combine the results of the encoder and sub-word predictor. Overall, our Conformer-Transducer-Small network has 9.4M parameters for the encoder, 0.9M parameters for the decoder, and 10.3M parameters in total.

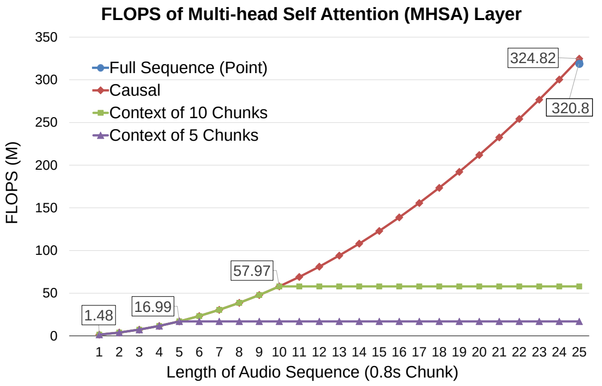


Figure 3: *FLOPS count for different masks of streaming Conformer. The causal model shows a quadratic increase in the computational cost as it processes the entire past sequence for every new chunk. Chunked-context mask based streaming Conformer-Transducer limits the computational cost by encoding a fixed number of chunks as context and decoding only the latest chunk.*

4.2. Training Settings:

We used LibriSpeech [29] 1000 hours dataset in our experiments. We trained the model on standard training-clean and training-other (960 hours) of the dataset and reported results for dev-clean and dev-other test sets. For pre-processing of the text data labels, we used sub-word conversion with a vocabulary size of 256. We trained each model for 200 epochs where each epoch took 25 minutes on 6 parallel NVIDIA A100 80G GPUs with a batch size of 4. We used AdamW [30] optimizer with its standard parameter settings. The networks trained with streaming masks take longer because these masks are defined dynamically during the training for different lengths of audio signals in the batch.

Table 1: *Word Error Rate (WER) of different modes of Conformer on LibriSpeech dataset.*

Computation Cost	Attention Mask	Word Error Rate (WER)	
		dev-clean	dev-other
Fixed	Non-Stream	5.52	13.6
Quadratic	Causal	5.79	14.61
Limited	Chunk	6.01	15.63
Limited (Ours)	5 Chunks	5.96	15.21
Limited (Ours)	10 Chunks	**5.81**	**14.65**

4.3. Results of Proposed Model:

The experimental results for Word Error Rate (WER) are reported in Table 1. First, we trained a non-streaming Conformer-Transducer network on full sequence length. This offline model produces the best result on the LibriSpeech dataset but is unable to perform ASR in a streaming way. We then trained the model with a causal mask which allows MHSA layers in Conformer to work for streaming ASR, but as it attends complete sequence from the past, its computational cost grows quadratically as shown in Figure 3. If we train Conformer with a chunk mask, allowing it to attend frames only in the current chunk, its WER performance degrades significantly. Our proposed chunked-context mask allows Conformer to maintain its performance by attending to a limited context and puts a cap on computational cost for long sequences by discarding the past frames in a chunked manner. We tried reducing the size of the context up to 5 chunks (including the current chunk), but the performance starts to degrade. Hence we find that keeping 10 chunks is the appropriate setting and delivered results closest to the causal network.

The computational cost of different modes of Conformer is computed in terms of Floating Point Operations (FLOPS) by using a Python package, DeepSpeed [31]. We mainly focused on the self-attention layer as it is the central layer of Transformer based models. The FLOPS for the MHSA module also involves linear layer computations for the key, query, value, and output vectors before computing self-attention. These linear layers have linear computational complexity with respect to the sequence length n, and quadratic complexity with respect to the dimension of the Conformer encoder d, represented as $O(nd^2)$. However, the attention operation has quadratic complexity with respect to the sequence length n and linear complexity with respect to the model dimension d, represented as $O(n^2d)$. Therefore, the overall FLOPS increase quadratically as the length of speech utterance increases, given that the model dimension is fixed. The trend of computational cost increase with respect to the utterance length is shown in Figure 3.

5. Real-time Deployment of Proposed Architecture

For the real-time performance analysis of our proposed architecture, we selected a single NVIDIA A100 GPU. We used an audio example from LibriSpeech dataset which is almost 15s long. We feed this audio to the network in form of streaming chunks. The length of each chunk is 0.8s which results in 18 total chunks. The number of context chunks n in this case is 10 including the current chunk. The proposed miniStreamer encoder processes the incoming stream with context chunks and the Transducer decodes the encodings into text labels only for the latest chunk. The y-axis represents Real Time Factor (RTF) which is the relation of end-to-end computational time and audio duration as represented by Equation 5.

$$RTF = \frac{time(decoder(encoder(x_{N-n}...x_N)))}{duration(x_N)} \quad (5)$$

Figure 4 shows that for the causal approach, RTF keeps increasing as the sequence length keeps growing, hence encoding and encoding time also increases. However, for the proposed chunked-context-based setting, the computational cost rises a little in the beginning and then stays constant as the new chunks keep arriving.

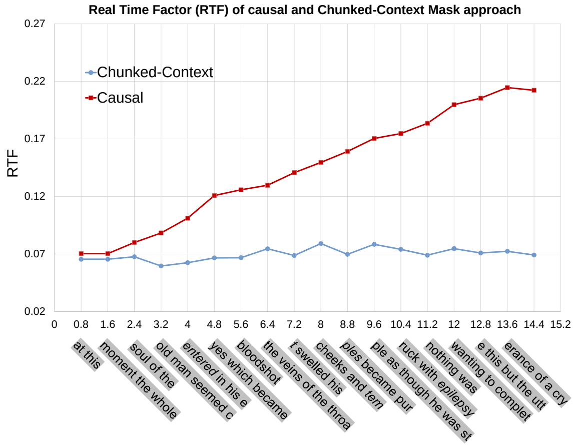


Figure 4: *The comparison between Real Time Factor (RTF) for conventional causal-based streaming approach and our proposed limited chunked-context masking approach. The x-axis shows the predicted text labels for each 0.8s audio chunk and y-axis shows the change in RTF with respect to time.*

The predicted labels for each chunk are shown along the horizontal axis as predicted labels of each chunk. It is clear that the Transducer decoder keeps a connection between sub-words for consecutive chunks. Therefore, our proposed approach can work for arbitrarily long sequences, without worrying about increasing computational cost for long sequences. 4.

6. Discussion

Our proposed architecture exploits self-attention in the Conformer and monotonic nature of ASR to optimally process the most relevant portion of the audio stream for streaming ASR. Several streaming proposals for Conformer report good accuracy, but it is important to highlight that the actual inference set-up for streaming Conformer has some challenges for shorter chunk lengths. Especially the normalization layers in Conformer cause deviation in predictions for small chunk sizes. In our case, 0.8s length worked reasonably well for chunked-based streaming ASR with Conformer-Small. With the proposed settings, miniStreamer achieved a constant RTF for streaming ASR of long audios, which is promising for real-time applications on the edge.

7. Conclusion

Conventional streaming techniques for Conformer cause a quadratic increase in the computational cost as the utterance length increases. In this work, we proposed a chunked-context-based streaming mask, which enhances Conformer-Transducer's capability to work for streaming ASR. Our proposed architecture, which combines chunk-based streaming Conformer with the Transducer decoder, resulted in significantly reducing the computational cost in terms of FLOPS while maintaining performance close to full-context causal streaming. The real-time inference analysis of the model shows that it maintains a low RTF even for long utterances when the conventional causal method results in an increasing RTF. The future work includes further improving the WER of miniStreamer and evaluating the model on a range of edge devices.

8. References

[1] N. Chivarov, D. Chikurtev, S. Chivarov, M. Pleva, S. Ondáš, J. Juhár, and K. Yovchev, "Case Study on Human-Robot Interaction of the Remote-Controlled Service Robot for Elderly and Disabled Care," *Computing and Informatics*, vol. 38, pp. 1210–1236, 01 2019.

[2] M. Tröbinger, C. Jähne, Z. Qu, J. Elsner, A. Reindl, S. Getz, T. Goll, B. Loinger, T. Loibl, C. Kugler, C. Calafell, M. Sabaghian, T. Ende, D. Wahrmann, S. Parusel, S. Haddadin, and S. Haddadin, "Introducing GARMI - A Service Robotics Platform to Support the Elderly at Home: Design Philosophy, System Overview and First Results," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5857–5864, 2021.

[3] J. Tang, S. Liu, L. Liu, B. Yu, and W. Shi, "LoPECS: A Low-Power Edge Computing System for Real-Time Autonomous Driving Services," *IEEE Access*, vol. 8, pp. 30 467–30 479, 2020.

[4] A. Martín García, I. Gonzalez-Carrasco, V. Rodriguez-Fernandez, M. Souto, D. Camacho, and B. Ruíz-Mezcua, "Deep-Sync: A novel deep learning-based tool for semantic-aware subtitling synchronisation," *Neural Computing and Applications*, pp. 1–15, 02 2021.

[5] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer," in *ASRU*, 12 2017, pp. 193–199.

[6] Y. He, R. Prabhavalkar, K. Rao, W. Li, A. Bakhtin, and I. McGraw, "Streaming small-footprint keyword spotting using sequence-to-sequence models," in *ASRU*, 12 2017, pp. 474–481.

[7] H. Sak, M. Shannon, K. Rao, and F. Beaufays, "Recurrent Neural Aligner: An Encoder-Decoder Neural Network Model for Sequence to Sequence Mapping," in *INTERSPEECH*, 08 2017, pp. 1298–1302.

[8] 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 *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7829–7833.

[9] J. Li, "Recent Advances in End-to-End Automatic Speech Recognition," *ArXiv*, vol. abs/2111.01690, 2021.

[10] 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 *INTERSPEECH*, 10 2020, pp. 5036–5040.

[11] T. N. Sainath, Y. He, A. Narayanan, R. Botros, W. Wang, D. Qiu, C.-C. Chiu, R. Prabhavalkar, A. Gruenstein, A. Gulati, B. Li, D. Rybach, E. Guzman, I. McGraw, J. Qin, K. Choromanski, Q. Liang, R. David, R. Pang, S.-Y. Chang, T. Strohman, W. R. Huang, W. Han, Y. Wu, and Y. Zhang, "Improving The Latency And Quality Of Cascaded Encoders," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 8112–8116.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *Neural Information Processing Systems (NIPS)*, 06 2017.

[13] S. Zhang, Z. Gao, H. Luo, M. Lei, J. Gao, Z. Yan, and L. Xie, "Streaming Chunk-Aware Multihead Attention for Online End-to-End Speech Recognition," in *INTERSPEECH*, 10 2020, pp. 2142–2146.

[14] D. Wang, Y. Shangguan, H. Yang, P. I.-J. Chuang, J. Zhou, M. Li, G. Venkatesh, O. Kalinli, and V. Chandra, "Noisy Training Improves E2E ASR for the Edge," *ArXiv*, vol. abs/2107.04677, 2021.

[15] R. Peinl, B. Rizk, and R. Szabad, "Open Source Speech Recognition on Edge Devices," in *10th International Conference on Advanced Computer Information Technologies (ACIT)*, 2020, pp. 441–445.

[16] G. Kurata and G. Saon, "Knowledge Distillation from Offline to Streaming RNN Transducer for End-to-End Speech Recognition," in *INTERSPEECH*, 2020.

[17] H. Yang, Y. Shangguan, D. Wang, M. Li, P. Chuang, X. Zhang, G. Venkatesh, O. Kalinli, and V. Chandra, "Omni-Sparsity DNN: Fast Sparsity Optimization for On-Device Streaming E2E ASR Via Supernet," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 8197–8201.

[18] A. Narayanan, R. Prabhavalkar, C.-C. Chiu, D. Rybach, T. N. Sainath, and T. Strohman, "Recognizing Long-Form Speech Using Streaming End-to-End Models," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 920–927.

[19] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi, J. Shi, S. Watanabe, K. Wei, W. Zhang, and Y. Zhang, "Recent Developments on ESPNet Toolkit Boosted By Conformer," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5874–5878.

[20] Z. Li, H. Miao, K. Deng, G. Cheng, S. Tian, T. Li, and Y. Yan, "Improving Streaming End-to-End ASR on Transformer-based Causal Models with Encoder States Revision Strategies," in *INTERSPEECH*, 09 2022, pp. 1671–1675.

[21] T. Hori, N. Moritz, C. Hori, and J. Le Roux, "Advanced Long-Context End-to-End Speech Recognition Using Context-Expanded Transformers," in *INTERSPEECH*, 08 2021, pp. 2097–2101.

[22] N. Moritz, T. Hori, and J. Le, "Streaming Automatic Speech Recognition with the Transformer Model," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6074–6078.

[23] B. X. Fangyuan Wang, Xiyuan Wang, "Sequentially Sampled Chunk Conformer for Streaming End-to-End ASR," vol. abs/2211.11419, 2022. [Online]. Available: https://arxiv.org/pdf/2211.11419.pdf

[24] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 369–376. [Online]. Available: https://doi.org/10.1145/1143844.1143891

[25] C.-C. Chiu and C. Raffel, "Monotonic Chunkwise Attention," *International Conference on Learning Representations (ICLR)*, 2018. [Online]. Available: https://openreview.net/pdf?id=Hko85plCW

[26] C. Gao, G. Cheng, R. Yang, H. Zhu, P. Zhang, and Y. Yan, "Pre-Training Transformer Decoder for End-to-End ASR Model with Unpaired Text Data," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6543–6547.

[27] F. Boyer, Y. Shinohara, T. Ishii, H. Inaguma, and S. Watanabe, "A Study of Transducer based End-to-End ASR with ESPnet: Architecture, Auxiliary Loss and Decoding Strategies," *CoRR*, vol. abs/2201.05420, 2022. [Online]. Available: https://arxiv.org/abs/2201.05420

[28] 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," in *INTERSPEECH*. ISCA, 2019, pp. 2613–2617.

[29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[30] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations (ICLR)*, 2017.

[31] GitHub Repository, "DeepSpeed," https://github.com/microsoft/DeepSpeed.