Shifted Chunk Encoder for Transformer Based Streaming End-to-End ASR

Fangyuan Wang[1(✉)] and Bo Xu[1,2,3]

[1] Institute of Automation, Chinese Academy of Science, Beijing, China
{fangyuan.wang,xubo}@ia.ac.cn
[2] School of Future Technology, University of Chinese Academy of Sciences, Beijing, China
[3] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China

Abstract. Currently, there are mainly three kinds of Transformer encoder based streaming End to End (E2E) Automatic Speech Recognition (ASR) approaches, namely time-restricted methods, chunk-wise methods, and memory-based methods. Generally, all of them have limitations in aspects of linear computational complexity, global context modeling, and parallel training. In this work, we aim to build a model to take all these three advantages for streaming Transformer ASR. Particularly, we propose a shifted chunk mechanism for the chunk-wise Transformer which provides cross-chunk connections between chunks. Therefore, the global context modeling ability of chunk-wise models can be significantly enhanced while all the original merits inherited. We integrate this scheme with the chunk-wise Transformer and Conformer, and identify them as SChunk-Transformer and SChunk-Conformer, respectively. Experiments on AISHELL-1 show that the SChunk-Transformer and SChunk-Conformer can respectively achieve CER 6.43% and 5.77%. And the linear complexity makes them possible to train with large batches and infer more efficiently. Our models can significantly outperform their conventional chunk-wise counterparts, while being competitive, with only 0.22 absolute CER drop, when compared with U2 which has quadratic complexity. A better CER can be achieved if compared with existing chunk-wise or memory-based methods, such as HS-DACS and MMA. Code is released. (see https://github.com/wangfangyuan/SChunk-Encoder.).

Keywords: Shifted Chunk Transformer · Shifted Chunk Conformer · Streaming ASR · Transformer · End-to-End ASR

1 Introduction

In the past decades, ASR with E2E models has achieved great progress, and has become a popular alternative to the hybrid ASR models equipped with conven-

This work is supported by the National Innovation 2030 Major S&T Project of China under Grant 2020AAA0104202 and the Key Research Program of the Chinese Academy of Sciences No. ZDBS-SSW-JSC006.

M. Tanveer et al. (Eds.): ICONIP 2022, CCIS 1792, pp. 39–50, 2023.
https://doi.org/10.1007/978-981-99-1642-9_4

tional Hidden Markov Model (HMM)/Deep Neural Network (DNN). Currently, Connectionist Temporal Classification (CTC) [1,2], Recurrent Neural Network Transducer (RNN-T) [3], and Attention based Encoder-Decoder (AED) [5,6] are the three mainstream E2E systems. Also, efforts to conduct performance comparisons [7] or the combination [8,9] of these models have been made. Recently, Transformer [10] has become a prevalent architecture, outperforming RNN [11] in AED systems [7]. Furthermore, Transformer can also use as an encoder with CTC [1] or Transducer [4]. And very recently, the Conformer [6] has been proposed which augments Transformer with convolution neural networks (CNN). Both Espnet [12] and WeNet [13] have shown that Conformer can bring significantly performance gains on a wide range of ASR corpora.

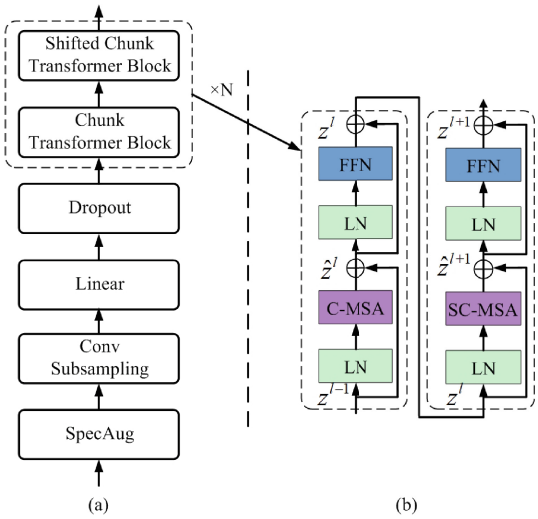

Fig. 1. (a) The architecture of SChunk-Transformer, N is set to 6 by default; (b) two successive blocks (notation presented with Eq. (3)).

The great success of Transformer and its variants urge people to explore its adaption for streaming ASR. However, two issues make vanilla models impractical for streaming ASR. First, the calculation of self-attention depends on the entire input sequence. Second, the computation and memory usage grow quadratically to the length of the input sequence. Actually, several methods have been proposed to alleviate these issues. 1) Time-restricted methods [14–17,27,28] where the attention computation only uses past input vectors and limited future inputs. However, the time and memory complexities of these methods are still quadratic, which may introduce a significant latency for long inputs. 2)Chunk-wise methods [4,18,26] typically evenly partition the input into chunks and then calculate attention only within these chunks as monotonic chunk-wise attention (MoChA) [19]. They have linear complexities but usually suffer dramatic performance drops as the reception field of attention is limited within local chunks.

3)Memory-based methods [20–22] utilize the solution of chunk-wise methods to reduce running time while employing an auxiliary contextual vector to memorize the history information. However, these vectors break the parallel nature of Transformer, typically requiring a longer training time.

In this paper, we aim to build a streaming Transformer which can compute in linear complexity, capture global history context and parallel train simultaneously. Under the guidance of this goal, we find inspiration from Swin Transformer [23] and introduce the idea of shifted windows into streaming ASR. In detail, we propose a shifted chunk mechanism for chunk-wise Transformer models. This mechanism allows the computation of attention to cross the boundary of chunks, thus can significantly enhance the model power, while keeping linear complexity and parallel training. We integrate the proposed mechanism into Transformer and Conformer and get SChunk-Transofromer and SChunk-Conformer, respectively. And we have conducted ablation studies and comparison experiments on AISHELL-1 [24]. The results show that Schunk-Transformer and Schunk-Conformer can respectively achieve CER 6.43% and 5.77% when set the chunk size to 16, which significantly surpass their conventional chunk-wise counterparts. When compared with U2 [16], which is a strong baseline model using the time-restricted method, our models can still be competitive with only an absolute 0.22 CER drop for SChunk-Conformer but be more efficient to train and infer. Superior performance can achieve if compared with other existing chunk-wise or memory-based methods, such as HS-DACS [26] and MMA [21].

2 Shifted Chunk Encoder

For convenience, we take SChunk-Transformer as an illustrative encoder to describe the mechanism of the shifted chunk.

2.1 Overall Architecture

As illustrated in Fig. 1(a), our proposed encoder first processes the input audios with SpecAug [25], convolution subsampling, and other frontend layers as conventional Transformer ASR, and then with several consecutive chunk Transformer blocks and shifted chunk Transformer blocks. The distinctive feature of our model is the use of chunk Transformer block and successively shifted chunk Transformer block to replace chunk Transformer blocks.

2.2 Shifted Chunk Transformer Block

We build the SChunk-Transformer block by replacing the multi-head self attention (MSA) in a Transformer block with a module based on shifted chunks (described in Sect. 2.3), with other layers kept the same, see Fig. 1(b). The SChunk-Transformer block is composed of a shifted chunk based MSA module, followed by a 2-layer Feed Forward Network (FFN) with GELU nonlinearity in between. It applies a LayerNorm (LN) layer before each MSA and FFN module and adds a residual connection after each module.

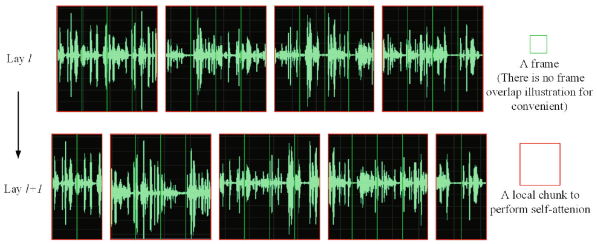

Fig. 2. An illustration of the shifted chunk approach for computing self-attention. In layer l (top), self-attention is computed in local chunks which are got by a regular chunk partitioning scheme. In the next layer $l+1$ (bottom), the self-attention computations are conducted in new chunks which cross the previous chunks in layer l and got by shifting.

2.3 Shifted Chunk Based Self-attention

Chunk-Wise Self-attention. The vanilla Transformer [10] uses global MSA to compute the dependencies between a frame and all the other frames. To be efficient, we calculate self-attention within evenly partitioned non-overlapped chunks. If an audio of L frames and each chunk has W frames, the complexities of computing a global MSA and a chunk based MSA are[1]:

$$\Omega(MSA) = 4L \cdot C^2 + 2L^2 \cdot C \tag{1}$$

$$\Omega(C\text{-}MSA) = 4L \cdot C^2 + 2N \cdot L \cdot C \tag{2}$$

where C is the feature dimension, the former is quadratic to L, and the latter is linear when W is a fixed value. Global MSA is generally unaffordable for a large L, which may introduce a significant latency for time-restricted methods.

Shifted Chunk Partitioning in Successive Blocks. The chunk based MSA lacks connections across chunks, which limits its modeling power. We propose the shifted chunk partition approach to introduce cross-chunk connections while maintaining the efficiency of chunk-wise computation. As shown in Fig. 2, we use the regular partitioned chunks followed by the shifted partitioned chunks consecutively. The regular chunk partitioning strategy starts from the audio, and the feature sequence of 16 frames is evenly partitioned into 4 chunks of size 4 ($W = 4$). Then, the shifted partition is shifted from the preceding layer, by displacing the chunks by $\lfloor W/2 \rfloor$ frames from the regularly partitioned chunks.

With the shifted chunk partitioning approach, the Chunk-Transformer block and SChunk-Transformer block are computed as:

[1] We omit softmax computation in determining complexity.

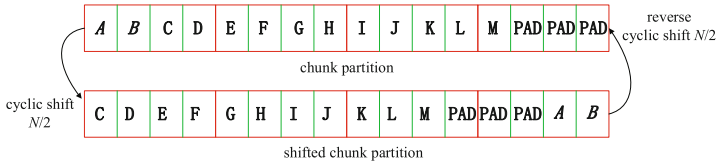

Fig. 3. An illustration of efficient batch computation for self-attention in shifted chunk partitioning.

$$\begin{aligned} \hat{z}^l &= C\text{-}MSA(LN(z^{l-1})) + z^{l-1}, \\ z^l &= FFN(LN(\hat{z}^l)) + \hat{z^l}, \\ \hat{z}^{l+1} &= SC\text{-}MSA(LN(z^l)) + z^l, \\ z^{l+1} &= FFN(LN(\hat{z}^{l+1})) + \hat{z}^{l+1} \end{aligned} \tag{3}$$

where $\hat{z}^l$ and z^l denote the outputs of the (S)C-MSA and the FFN for block l, respectively; S-MSA and SC-MSA denote chunk based multi-head self attention using regular and shifted chunk partitioning configurations, respectively.

Efficient Batch Computation for Shifted Chunks. The first issue of shifted chunk partitioning for batch computation is the difference in audio lengths. To be evenly partitioned, we pad audios in a batch to the same length, which is a little longer than the longest one in the batch while can be evenly divided by the chunk size. Another issue is that shifted chunk partitioning will result in more chunks, and some chunks will be smaller than W, see Fig. 2. We use a batch computation approach by cyclic-shifting the regular partitioned chunks from head to tail to get the shifted partitioned chunks, and reverse cyclic-shifting the shifted partitioned chunks from tail to head to re-get the regular partitioned chunks, see Fig. 3. With the cyclic-shift, the number of batched chunks remains the same as that of regular chunk partitioning, and thus is also efficient.

Shifted Chunk Attention Mask. As shown in Fig. 4(a), the chunk based self-attention can compute using a chunk-wise attention mask to support streaming. However, for the shifted chunks, we need to mask out some areas as shown in Fig. 4(b) to make sure frames can only attend to their preceding ones when calculating the chunk-wise attention of SC-MSA.

3 Streaming ASR with Shifted Chunks

3.1 Streaming Encoder and Decoder

The SChunk-Transformer equipped with an attention mask can also support the streaming process as other chunk-wise methods. The casual convolution is used in SChunk-Conformer to make the CNN modules support streaming as in [16].

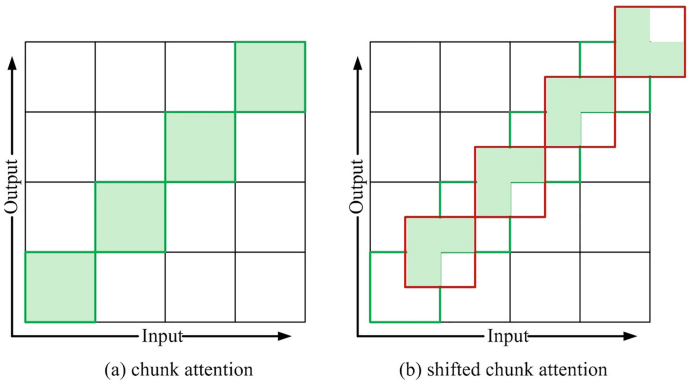

Fig. 4. Illustration of masks for chunk attention and shifted chunk attention.

We generally follow the decoder of U2 [16] that uses a hybrid CTC/Attention decoder. The CTC decoder outputs the first pass hypotheses in a streaming way. And then, the Attention decoder outputs the final results using full context to rescore the first pass hypotheses.

3.2 Streaming Inference

In the inference stage, the encoder consumes the inputs chunk by chunk. There is no shift in SC-MSA for the first chunk, and it degrades to behavior as C-MSA in this case without any impact on the first word prediction. For the subsequent chunks, we need to cache the past chunks and concatenate them with the current chunk as the input for the encoder, like the time-restricted methods. Once the CTC decoder receives the output of the encoder, it generates output immediately. At the end of an utterance, the Attention decoder is triggered to re-score the output of the CTC decoder to get a better utterance level result.

4 Experiments

4.1 Data

We evaluate the proposed models on AISHELL-1 [24], which contains 150 h of the training set, 10 h of dev set and 5 h test set, the test set consists of 7176 utterances in total. The official vocabulary contains 4233 tokens.

4.2 Experimental Setup

We implement models using the WeNet toolkit [13] and verify on two NVIDIA Gefore RTX 3090 GPUs (24G). For most hyper-parameters, we follow the recipes of WeNet. (FBank) splice 3-dimensional pitch computed on 25 ms window with

Table 1. Comparisons with different chunk size (CER%)

Model Architecture	# Chunk Size			
	4	8	16	32
Chunk-Transformer	31.30	18.86	11.80	7.66
Chunk-Conformer	6.55	6.33	6.09	5.90
SChunk-Transformer	7.76	6.68	6.43	5.92
SChunk-Conformer	6.74	6.21	5.77	5.64

10ms shift as input feature. And speed perturbation with 0.9, 1.0, and 1.1 are done to get 3-fold data. SpecAug [25] is applied with 2 frequency masks with a maximum frequency mask (F = 50), and 2-time masks with a maximum time mask (T = 50). Two convolution sub-sampling layers with kernel size 3×3 and stride 2 are used as the frontend. A stack of 4 heads SChunk-Transformer or SChunk-Conformer layers (12 by default) is used as the encoder. We use a CTC decoder and an Attention decoder of 6 transformer layers with 4 heads. The attention dimension is 256 and the feed forward dimension is 2048. Accumulating grad is used to stabilize training which updates every 4 steps. Attention dropout, feed forward dropout, and label smoothing regularization are applied in each encoder and decoder layer to prevent over-fitting. We use the Adam optimizer with the peak learning rate of 0.002 and transformer schedule to train these models for 80 epochs (batch size and warm-up steps are decided based on the memory usage of a model, set to 40 and 25000 by default). And get the final model by averaging the top 20 best models with the lowest loss on the dev set in the training stage.

4.3 Baseline Systems

Chunk-Transformer. We take the Chunk-Transformer and Chunk-Conformer, which we implemented using WeNet, as the first baseline models. The only difference between them and the proposed models is whether the shifted chunk mechanism is used or not.

U2. We take U2 [16], a built-in solution in WeNet, as a strong baseline since it's a SOTA model of the time-restricted methods and our models use the same decoder.

4.4 Ablation Studies

Chunk Size. First, we explore how chunk size affects performance. As shown in Table 1, we can see that better CERs can be achieved as the chunk size gets larger for both SChunk-Transformer and SChunk-Conformer. This implies large chunk size is beneficial to capture more global context. However, we need to balance the accuracy and latency and set the size to 16 for the following experiments.

Table 2. Comparisons with different number of encoder layers (CER%)

Model Architecture	# Encoder Layers			
	12	14	16	18
SChunk-Transformer	6.43	6.25	6.02	6.12
SChunk-Conformer	5.77	5.81	5.98	6.72

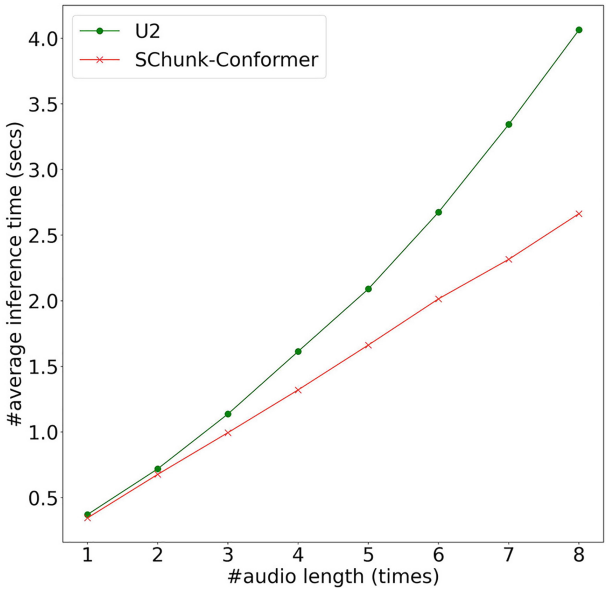

Fig. 5. The illustration of inference time cost of U2 and SChunk-Conformer. We concatenate each audio with itself several times in the test set of AISHELL-1 to imitate different audio lengths. All the inferences conducted on CPU (Intel(R) Xeon(R) Silver 4210R CPU @ 2.40GHz) with 1-thread, the y-axis indicates the average inference time of 7176 audios.

#Encoder Layers. We also investigate using more encoder layers to allow sufficient global context capturing. The results are shown in Table 2, the SChunk-Transformer achieves the best CER with 16 layers, while SChunk-Conformer achieves the best CER using 12 layers. We conjecture this is because the complicated encoder is easier to overfit. We set the encoder layer to 12 by default to make our models have similar parameters to others.

4.5 Comparisons with Baseline Systems

Chunk-Transformer: As shown in Table 1, the CER of Chunk-Conformer is significantly improved compared with Chunk-Transformer, the reason is attributed to the use of CNN to capture sequential history information. With

Table 3. Comparisons with U2. max. batch (#) is the maximum batch size each model can support on two RTX 3090 GPUs, training time is the total time cost of models trained for 80 epochs with each maximum bath size.

Model Architecture	max. batch (#)	trn. time (h)	CER (%)
U2 (static) [16]	48	29.13	5.55
U2 (dynamic) [16]	48	30.56	5.42
SChunk-Conformer	60	21.58	5.77

Table 4. Comparisons with other streaming solutions (CER%), ♮,†, and ‡ indicate the solution is a time-restricted method, a chunk-wise method and a memory-based method, respectively. Following [16], the latency is defined as the chunk size plus the right context (if any). △ is the additional latency introduced by rescoring.

Model Architecture	Type	Time Complexity	Latency(ms)	LM	CER(%)
Sync-Transformer [18]	†	linear	400		8.91
SCAMA [20]	‡	linear	600		7.39
MMA-narrow [21]	‡	linear	960		7.50
MMA-wide [21]	‡	linear	1920		6.60
HS-DACS [26]	†	linear	1280		6.80
SChunk-Transformer(ours)	†	linear	640+△		**6.43**
U2++ (U2+BiDecoding) [17]	♮	quadratic	640+△		5.05
WNARS(w/ rescoring) [27]	♮	quadratic	640+△	√	5.22
CUSIDE [28]	♮	quadratic	400+2		5.47
CUSIDE(w/NNLM rescoring) [28]	♮	quadratic	400+2	√	4.79
SChunk-Conformer(ours)	†	linear	640+△		**5.77**

the shifted chunk mechanism, our SChunk-Transformer can also significantly improve the CER of Chunk-Transformer, which verifies the proposed mechanism can help enhance the ability to model global context. The comparison of SChunk-Conformer and Chunk-Conformer confirms the phenomenon with an exception when the chunk size is 4. This may be because the shifted chunks with attention mask cannot use the whole chunk to model will bring a negative impact in the case of extremely small chunk size.

U2: As a strong baseline, U2 can achieve slightly better CER compared with our SChunk-Conformer, see Table 3. This indicates that the time-restricted methods using full context are beneficial to get better accuracy. However, the performance gap between the SChunk-Conformer and U2 (static, train using static chunk size [16]) is quite narrow, with only 0.22 absolute CER drop. On the other hand, our models can use a much larger batch size to train, maxium batch size is 60 for SChunk-Conformer while 48 for U2, which can significantly reduce the training time as shown in Table 3. And the average inference time of SChunk-Conformer is linear to the audio length while quadratic for U2, see Fig. 5, which is important

to control system latency for streaming ASR. All in all, compared with U2, our models not only can achieve competitive CER, but also can train and infer more efficiently.

4.6 Comparisons with Other Streaming Solutions

Table 4 lists several recently published Transformer based streaming solutions. We can see that the SChunk-Transformer can surpass all the chunk-wise or memory-based models, with 0.37 and 0.17 absolute CER improvement compared with HS-DACS [26] and MMA [21], respectively. Compared with the other time-restricted models, which use sophisticated techniques (for example, language model (LM) rescoring) to further boost performance compared with U2 [16], it's not surprise that our SChunk-Conformer fails to achieve superior CER as a chunk-wise model. However, either U2 or other advanced time-restricted models all have qudratic complexity, in contrast SChunk-Conformer can train and infer more efficiently which is crucial for streaming ASR.

Compared with the other advanced time-restricted models [17,27,28], which use sophisticated techniques to further boost performance compared with [16], it's no surprise that our SChunk-Conformer fails to achieve superior CER as a chunk-wise model. However, either U2 or other advanced time-restricted models all have quadratic time and memory complexities, in contrast, Schunk-Conformer has linear complexity and can train and infer more efficiently which is crucial for streaming ASR.

5 Discussion

Our work shows a way to build a single streaming E2E ASR model to achieve the benefits of linear complexity, global context modeling, and parallel trainable concurrently. Despite the time-restrict models can achieve slightly better CERs, they cannot ensure a low latency in theory makes them impractical for scenarios with long audios. As the shifted chunk based models can achieve competitive CERs while be insensitive to audio length, they may have a great potential in commercial systems.

6 Conclusions

We introduce a shifted chunk mechanism for chunk-wise Transformer and Conformer models. This mechanism can significantly enhance the modeling power by allowing local self-attention to capture global context across chunks while keeping linear complexity and parallel trainable. Experimental results on AISHELL-1 show that both SChunk-Transformer and SChunk-Conformer can significantly outperform Chunk-Transformer and Chunk-Conformer, respectively. And, the SChunk-Transformer can surpass the SOTA models of both chunk-wise methods and memory-based methods. Compared with the time-restricted methods, our SChunk-Conformer can achieve competitive CER while being able to train and infer more efficiently. In the future, we plan to pay more attention to exploring effective cross-chunk self-attention modeling methods to further improve the performance of streaming ASR.

References

1. Li, J., Ye, G., Das, A., Zhao, R., Gong, Y.: Advancing acoustic-to-word CTC model. In: ICASSP 2018–43rd IEEE International Conference on Acoustics, Speech and Signal Processing, 22–27 April, Seoul, South Korea, pp. 5794–5798 (2018)
2. Graves, A., Fernandez, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: ICML 2006–23rd International Conference on Machine Learning, 25–29 June, Pittsburgh, Pennsylvania, pp. 369–376 (2006)
3. Battenberg, E., Chen, J.T., et al.: Exploring neural transducers for end-to-end speech recognition. In: ASRU 2017–2017 IEEE Automatic Speech Recognition and Understanding Workshop, 16–20 December, Okinawa, Japan, pp. 206–213 (2017)
4. Chen, X., Wu, Y., Wang, Z., et al.: Developing real-time streaming transformer transducer for speech recognition on large-scale dataset. In: ICASSP 2021–46rd IEEE International Conference on Acoustics, Speech and Signal Processing, 6–11 June, Toronto, Ontario, Canada, pp. 5904–5908 (2021)
5. Chan, W., Jaitly, N., Le, Q., Vinyals, O.: Listen, attend and spell: a neural network for large vocabulary conversational speech recognition. In: ICASSP 2016–41rd IEEE International Conference on Acoustics, Speech and Signal Processing, 20–25 March, Shanghai, China, pp. 4960–4964 (2016)
6. Gulati, A., Qin, J., Chiu, C.C., et al.: Conformer: convolution-augmented transformer for speech recognition. In: Interspeech 2020–21rd Annual Conference of the International Speech Communication Association, 25–30 October, Shanghai, China, pp. 5036–5040 (2020)
7. Prabhavalkar, R., Rao, K., Sainath, T.N., Li, B., Johnson, L., Jaitly, N.: A comparison of sequence-to-sequence models for speech recognition. In: Interspeech 2017–18rd Annual Conference of the International Speech Communication Association, 20–24 August, Stockholm, Stockholm County, Swedenm, pp. 939–939 (2017)
8. Watanabe, S., Hori, T., Kim, S., Hershey, J.R., Hayashi, T.: Hybrid CTC/Attention architecture for end-to-end speech recognition. IEEE J. Sel. Top. Sign. Process. **11**(8), 1240–1253 (2017)
9. Miao, H.R., Cheng, G.F., Zhang, P.Y., Yan, Y.H.: Online Hybrid CTC/Attention end-to-end automatic speech recognition architecture. IEEE/ACM Trans. Audio Speech Lang. Process. **28**, 1452–1465 (2020)
10. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. In: NIPS 2017–31rd Conference on Neural Information Processing Systems, 4–9 December, Long Beach, California, U.S.A., pp. 5998–6008 (2017)
11. Zhao, Y., Zhou, S., Xu, S., Xu, B.: Word-level permutation and improved lower frame rate for rnn-based acoustic modeling. In: ICONIP 2017–24rd International Conference on Neural Information Processing, 14–18 November, Guangzhou, China (2017)
12. Guo, P.C., Boyer, F., Chang, X.K., et al.: Recent developments on Espnet toolkit boosted by conformer. In: ICASSP 2021–46rd IEEE International Conference on Acoustics, Speech and Signal Processing, 6–11 June, Toronto, Ontario, Canada, pp. 5874–5878 (2021)
13. Yao, Z., Wu, D., Wang, X., et al.: WeNet: production oriented streaming and non-streaming end-to-end speech recognition toolkit. In: Interspeech 2021–22rd Annual Conference of the International Speech Communication Association, 30 August-3 September, Brno, Czech Republic (2021)

14. Yu, J.H., Han, W., et al.: Universal ASR: unify and improve streaming ASR with full-context modeling. arXiv preprint arXiv:2010.06030 (2020)
15. Tripathi, A., Kim, J., Zhang, Q., et al.: Transformer transducer: one model unifying streaming and non-streaming speech recognition. arXiv preprint arXiv:2010.03192 (2020)
16. Zhang, B.B., Wu, D., Yao, Z.Y., et al.: Unified streaming and non-streaming two-pass end-to-end model for speech recognition. arXiv preprint arXiv:2012.05481 (2020)
17. Wu, D., Zhang, B.B., Yang, C., et al.: U2++: unified two-pass bidirectional end-to-end model for speech recognition. arXiv preprint arXiv:2106.05642 (2021)
18. Tian, Z.K., Yi, J.Y., Bai, Y., et al.: Synchronous transformers for end-to-end speech recognition. In: ICASSP 2020–45rd IEEE International Conference on Acoustics, Speech and Signal Processing, 4–8 May, Barcelona, Spain, pp. 7884–7888 (2020)
19. Chiu, C.-C., Raffel, C.: Monotonic chunkwise attention. In: ICLR 2018–6rd International Conference on Learning Representations, 30 April-3 May, Vancouver Canada (2018)
20. Zhang, S.L., Gao, Z.F., Luo, H.N., et al.: Streaming chunk-aware multihead attention for online end-to-end speech recognition. In: Interspeech 2020–21rd Annual Conference of the International Speech Communication Association, 25–30 October, Shanghai, China, pp. 2142–2146 (2020)
21. Inaguma, H., Mimura, M., Kawahara, T.: Enhancing monotonic multihead attention for streaming ASR. In: Interspeech 2020–21rd Annual Conference of the International Speech Communication Association, 25–30 October, Shanghai, China, pp. 2137–2141 (2020)
22. Shi, Y.Y., Wang, Y.Q., Wu, C.Y., et al.: Emformer: efficient memory transformer based acoustic model for low latency streaming speech recognition. In: ICASSP 2021–46rd IEEE International Conference on Acoustics, Speech and Signal Processing, 6–11 June, Toronto, Ontario, Canada, pp. 6783–6787 (2021)
23. Liu, Z., Cao, Y.T., et al.: Swin transformer: hierarchical vision transformer using shifted windows. In: ICCV 2021–46rd International Conference on Computer Vision, 11–17 October, Virtual, pp. 10012–10022 (2021)
24. Bu, H., Du, J., Na, X., Wu, B., Zheng, H.: Aishell-1: an open-source mandarin speech corpus and a speech recognition baseline. In: O-COCOSDA 2017–20rd Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment, 1–3 November, Seoul, South Korea, pp. 1–5 (2015)
25. Park, D.S., Chan, W., Zhang, Y., Chiu, C.C., et, al.: Specaugment: a simple data augmentation method for automatic speech recognition. In: Interspeech 2019–20rd Annual Conference of the International Speech Communication Association, 15–19 September, Graz, Austria, pp. 2613–2617 (2019)
26. Li, M., Zorilă, C., Doddipatla, R.: Head-synchronous decoding for transformer-based streaming ASR. In: ICASSP 2021–46rd IEEE International Conference on Acoustics, Speech and Signal Processing, 6–11 June, Toronto, Ontario, Canada, pp. 5909–5913 (2021)
27. Wang, Z., Yang, W., Zhou, P., Chen, W.: WNARS: WFST based non-autoregressive streaming end-to-end speech recognition. arXiv preprint arXiv:2104.03587 (2021)
28. An, K., Zheng, H., Ou, Z., Xiang, H., Ding, K., Wan, G.: CUSIDE: chunking, simulating future context and decoding for streaming ASR. arXiv preprint arXiv:2203.16758(2022)