U2++: Unified Two-pass Bidirectional End-to-end Model for Speech Recognition

Di Wu[1]*, Binbin Zhang*[1]*, Chao Yang*[1]*, Zhendong Peng*[1]*, Wenjing Xia*[1]*, Xiaoyu Chen*[1]*, Xin Lei*[1]

[1]Mobvoi Inc., Beijing, China

di.wu@mobvoi.com binbinzhang@mobvoi.com chaoyang@mobvoi.com zhendong.peng@mobvoi.com
wenjing.xia@mobvoi.com xiaoyu.chen@mobvoi.com mikelei@mobvoi.com

Abstract

The unified streaming and non-streaming two-pass (U2) end-to-end model for speech recognition has shown great performance in terms of streaming capability, accuracy, real-time factor (RTF), and latency. In this paper, we present U2++, an enhanced version of U2 to further improve the accuracy. The core idea of U2++ is to use the forward and the backward information of the labeling sequences at the same time at training to learn richer information, and combine the forward and backward prediction at decoding to give more accurate recognition results. We also proposed a new data augmentation method called SpecSub to help the U2++ model to be more accurate and robust. Our experiments show that, compared with U2, U2++ shows faster convergence at training, better robustness to the decoding method, as well as consistent 5% - 10% word error rate reduction gain over U2. On the experiment of AISHELL-1, we achieve a 4.63% character error rate (CER) with a non-streaming setup and 5.05% with a streaming setup with 320ms latency by U2++. To the best of our knowledge, 5.05% is the best-published streaming result on the AISHELL-1 test set.

Index Terms: speech recognition, bidirectional re-score, WeNet, U2++

1. Introduction

End-to-end (E2E) models such as CTC[1, 2], recurrent neural network transducer (RNN-T)[3, 4], and attention-based encoder-decoder (AED)[5, 6, 7] gained more and more attention over the last few years. Compared with the conventional hybrid ASR framework, E2E models not only extremely simplified training and decoding procedure but also show superior performance in the standard of word error rate (WER). So applying the E2E models into real-world productions becomes necessary. However, deploying an E2E system is not trivial and there are a lot of practical problems to be solved.

The first is the streaming problem, some state-of-the-art models such as vanilla AED could not run in a streaming way, which limits the application scenarios. The second is that streaming and non-streaming systems are usually developed, trained, and deployed separately, which is resource-consuming. The third is the production problem, a lot of engineer efforts are required to promote e2e models to the production level.

Our previously proposed U2[8], a unified streaming and non-streaming end-to-end model, and WeNet[**?**], a production first and production-ready E2E toolkit solve the above problems in a simple and graceful way.

In this paper, we present U2++, an enhanced version of U2 to further improve the performance. Compared with U2, where there is only one left-to-right (forward) decoder, we add another right-to-left (backward) decoder. It means we have bidirectional decoders in U2++, which uses both the forward and the backward information of the labeling sequence. In the training, the model is trained jointly by CTC loss, left-to-right AED loss, right-to-left AED loss, so it learns richer information than U2. In the decoding, both the left-to-right decoder and the right-to-left decoder are used for re-scoring the n-best hypotheses from the CTC decoder, so we can make a more accurate prediction. Besides, inspired by SpecSwap[10] and SpecAugment[11], we also proposed a novel data augmentation method called SpecSub, which randomly and dynamically replace the current feature block with the feature block before. Our experiments show that U2++ is more stable, accurate, and robust than U2. The CTC and bidirectional AED joint training benefits both the CTC result and the final re-scoring result. On average, we get a 5%-8% relative word error rate reduction by U2++. On the AISHELL-1 dataset[12], we get a 5.05% CER on the test set with a streaming setting. As far as we know, it is the best streaming result on AISHELL-1.

2. Related Works

Hybrid CTC/attention end-to-end ASR[13] adopts both CTC and attention decoder loss during training, which results in faster convergence and also improves the robustness of the AED model. The encoder module can use any neural network which can model contexts, such as CNN, LSTM, conventional Transformer encoder, and recently proposed Conformer encoder which gets SOTA results on many public data sets. The decoder module of the hybrid CTC/attention model usually uses a conventional transformer decoder, in which left-to-right self-attention and across attention are used.

There are many decoding strategies of the hybrid CTC/attention model during the decoding process. EspNet[13] combines CTC search and auto-regressive beam search decoding of the attention decoder. U2 adopts a two-pass decoding strategy, the first pass is a CTC prefix beam search which gives the n-best candidate hypothesizes by CTC decoder, the second pass is a re-scoring process that re-scores the n-best candidates and the candidate with the best score is chosen to be the final result.

There are some works on bidirectional decoder on conventional speech transformer to improve the accuracy. [14] adds a right-to-left decoder to the conventional speech transformer model, and both the left-to-right decoder and the right-to-left decoder use auto-regressive beam search during the decoding. [15] use a BERT style decoder instead of the conventional speech transformer so that it models both the left and the right context, but the decoding process is complicated. Pika[1], a RNN-T based E2E toolkit, supports bidirectional LAS rescoring on RNN-T model, however, experimental results, experi-

[1]https://github.com/tencent-ailab/pika

mental analysis and recipes are not given.

3. U2++

3.1. Model architecture

The proposed model architecture is shown in Figure1. It contains four parts, a *Shared Encoder* that models the context of the acoustic features, a *CTC Decoder* that models the alignment of the frames and tokens, a *Left-to-Right Attention Decoder (L2R)* that models the left tokens dependency, and a *Right-to-Left Attention Decoder (R2L)* model the right tokens dependency. The *Shared Encoder* consists of multiple Transformer[16] or Conformer[17] encoder layers. The *CTC Decoder* consists of a linear layer and a log softmax layer. The CTC loss function is applied over the softmax output in training.

The *Left-to-Right Attention Decoder* and the *Right-to-Left Attention Decoder* use the same model structure as the conventional transformer decoder. We still make the *Shared Encoder* only see limited right context which is the same as U2, then the CTC decoder could run in a streaming mode in the first pass. In the second pass, we still use the re-scoring method as used in U2, but since we have two decoders in U2++, the re-scoring process will be a little different from U2. The detailed training and decoding processes are described in the following.

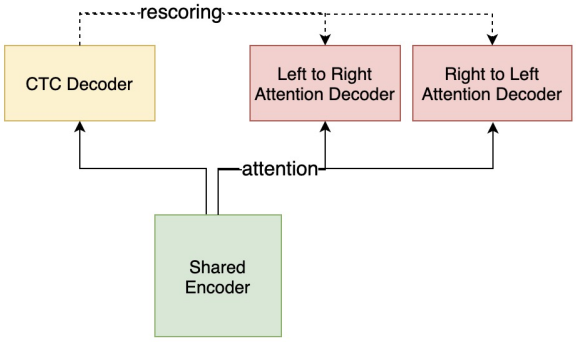

Figure 1: *Two pass CTC and AED joint architecture*

3.2. Training

3.2.1. Combined Loss

The training loss is the combined CTC and AED loss as listed in the equation1, and the AED loss consists of an L2R AED loss and an R2L AED loss as listed in the equation 2. Both of the two attention decoders are trained in the teacher forcing mode. In both equation 2 and equation 1, $\mathbf{x}$ are the acoustic features, $\mathbf{y}$ are the corresponding annotations, $\mathbf{L}_{\text{CTC}}(\mathbf{x}, \mathbf{y})$ and $\mathbf{L}_{\text{AED}}(\mathbf{x}, \mathbf{y})$ are the CTC and AED loss respectively. λ is a hyperparameter that balances the importance of the CTC and AED loss. α is another hyperparameter that balances the R2L AED loss and the L2R AED loss.

$$\mathbf{L}_{\text{combined}}(\mathbf{x}, \mathbf{y}) = \lambda \mathbf{L}_{\text{CTC}}(\mathbf{x}, \mathbf{y}) + (1-\lambda)(\mathbf{L}_{\text{AED}}(\mathbf{x}, \mathbf{y})) \quad (1)$$

$$\mathbf{L}_{\text{AED}}(\mathbf{x}, \mathbf{y}) = (1-\alpha)\mathbf{L}_{\text{AED-L2R}}(\mathbf{x}, \mathbf{y}) + \alpha(\mathbf{L}_{\text{AED-R2L}}(\mathbf{x}, \mathbf{y})) \quad (2)$$

3.2.2. Bi-directional Attention Decoder

In [14], the L2R and R2L decoders share the same weights. In the decoding process, these two decoders run separately and the sequence with the best score is chosen as the final result. Although This strategy could give more candidates, it is hard to combine the two direction scores due to the autoregressive mode.

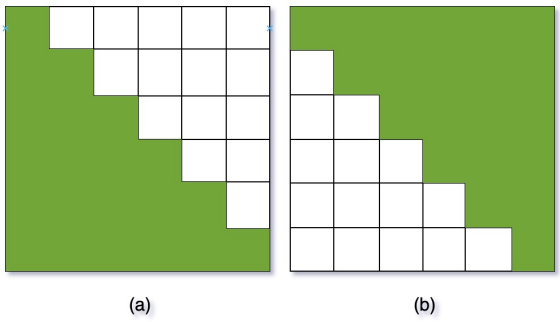

Figure 2: *left attention mask and right attention mask*

Our models have mainly two differences from [14] . First, we used different weights for the two decoders. Because the reversed version language should not be considered as the reasonable original language. Second, in the U2++ framework, the attention decoders are used for re-scoring instead of autoregressive decoding, so the two decoder scores of the candidates could be combined.

The implementations are described in the following. The mask in Figure 2(a) is applied to the self-attention in the L2R decoder to make each word only see its left words. Similarly, The mask in Figure 2(b) is applied in the R2L decoder to make each word only see its right words. Another implementation is to reverse the input sentence and use Figure 2(a) mask. The two implementations are only different in the position encoding information and give almost the same performance in experiments. So in this paper, we only report the result of the first implementation.

3.2.3. Spectral Augmentation

We propose a new spectral augmentation method called SpecSub to make the model more robust. We randomly substitute a frame chunk with the previous chunk. This simple augmentation shows significant performance improvements in the experiments.

Algorithm 1 SpecSub Augmentation

Initialization: $T_{max}, T_{min}, N_{max}, n = 1$
sample $N \sim Uni(0, N_{max})$
for n from 1 to N **do**
  sample $\Delta_t \sim Uni(T_{min}, T_{max})$
  sample $t \sim Uni(0, T - \Delta_t)$
  sample $t' \sim Uni(0, t)$
  $f(t, t + \Delta_t) \leftarrow f(t', t' + \Delta_t)$
end for

3.3. Decoding

We use the similar two-pass re-scoring decoding strategy with U2, while the new R2L decoder score is involved in re-scoring in U2++.

In the first pass stage, the frame-sync CTC decoder outputs the candidates using prefix beam search. In the second pass stage, the two attention decoder scores of these n-best candidates are calculated. The final score is calculated via equation

3. The candidate with the best score is chosen as the final result.

$$\mathbf{S}_{\text{final}} = \lambda \times \mathbf{S}_{\text{CTC}} + (1-\alpha) \times \mathbf{S}_{\text{L2R}} + \alpha \times \mathbf{S}_{\text{R2L}} \quad (3)$$

4. Experiments

To evaluate the proposed U2++ model, we carry out our experiments on the open-source Chinese Mandarin speech corpus AISHELL-1[12] and the open-source Chinese Mandarin speech corpus AISHELL-2. AISHELL-1 contains a 150-hour training set, a 20-hour development set, and a 10-hour test set. The test set contains 7176 utterances in total. AISHELL-2 contains a 1000-hour training set, and we use dev_ios as a development set and $test\_ios$ that contains 5000 utterances in total as the test set for validation and evaluation. We use WeNet[2] - an end-to-end speech recognition toolkit for all experiments.

The 80-dimensional log Mel-filter banks (FBANK) computed on-the-fly by torchaudio with a 25ms window and a 10ms shift are used as the features. SpecAugment [11] is applied with two frequency masks with maximum frequency mask ($F = 10$), and two times masks with maximum time mask ($T = 50$) to alleviate the over-fitting. The T_{max}, T_{min}, N_{max} of the proposed SpecSub is set to 30, 0 and 3 respectively.

We use two state-of-the-art ASR networks Conformer and Transformer as our shared encoder to evaluate U2++. We replace the batch-norm in the convolution module of Conformer with layer-norm, and we use the causal convolution in the convolution module. The relative shift in the original Conformer is removed for streaming. Two convolution sub-sampling layers with kernel size 3*3 and stride 2 are used in the front of the encoder. The kernel size of the convolution is 8 in Conformer. We use 12 transformer/conformer layers for the encoder. 3 transformer layers are used for both the L2R decoder and the R2L decoder, and the number of attention heads, attention dimension, and feed-forward dimension are set 4, 256, and 2048 respectively. So the parameters in U2++ are almost the same as U2.

Adam optimizer is used and the learning rate is warmed up with 25,000 steps. The α and λ are both set to 0.3 in training. In decoding, α and λ are set according to the developing set performance. In AISHELL-1 task, they are 0.3 and 0.5. In AISHELL-2 task, they are 0.1 and 0.5. The final model averages the top 30 models with the best validation loss on the development set. The beam size of the CTC prefix search is ten.

Table 1: *Test set CER (%) of U2/U2++ trained on AISHELL-1 Mandarin speech data.*

training method	decoding mode	decoding chunk size	
		full	16
U2 Transformer	ctc prefix beam search	6.28	6.98
	attention rescoring	5.52	6.05
U2++ Transformer	ctc prefix beam search	6.05	6.92
	attention rescoring	5.09	5.66
U2 Conformer	ctc prefix beam search	5.57	6.30
	attention rescoring	4.97	5.45
U2++ Conformer	ctc prefix beam search	5.19	5.81
	attention rescoring	4.63	5.05

4.1. AISEHLL-1 Task

We use the same dynamic chunk training as U2, which enables the model to work well in different chunk sizes (full and 16). The full chunk is for a non-streaming manner and chunk=16 is for streaming. the model latency of chunk=16 mode is ranged from 0 to 640ms, so the average latency is 320ms. Speed perturbation with 0.9, 1.0, and 1.1 is applied on the fly on the whole data.

[2]https://github.com/mobvoi/wenet

Table 2: *Comparison to other streaming solutions on AISHELL-1 task*

model	latency(ms)	lm	CER
Sync-Transformer[18]	400	n	8.91
SCAMA[19]	600	n	7.39
MMA[20]	640	n	6.60
U2 [8]	**320**+Δ	n	**5.33**
WNARS [21]	640	y	**5.15**
U2++	**320**+Δ	n	**5.05**

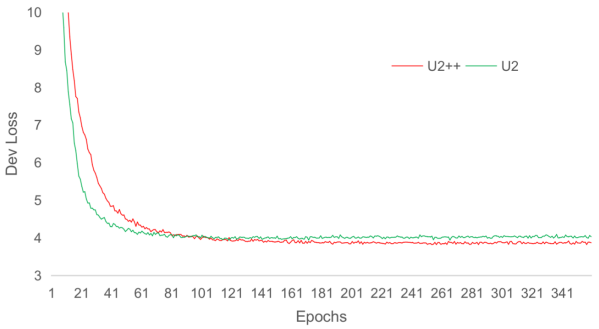

Figure 3: *The loss comparison of U2++ and U2*

The model performance comparisons of U2 and U2++ are shown in Table 1, U2++ is better than U2 on both Transformer and Conformer model in both streaming and non-streaming mode. The U2++ and U2 Conformer model losses are shown in Figure 3. The loss of U2++ is higher than U2 in the early stage, but it converges to lower after enough epochs.

Table 2 shows the comparisons with the other published systems on this task. 'y' in 'lm' column means a separated language model is used and 'n' means not. The averaged latency of the U2 and U2++ model is about 320ms and Δ usually is about 100ms. U2++ also achieves the best CER results in streaming mode on the AISHELL-1 data set and even better than WNARS which uses a separated LM during the decoding process.

4.2. AISEHLL-2 Task

We extend our experiments on the larger AISHELL-2 dataset. In AISHELL-2, speed perturbations are not applied. The performance of U2++ is listed in Table 3. On this dataset, U2++ still does better than U2 both on the Transformer and the Conformer model.

Table 3: *AISHELL-2 Mandarin speech data trained U2 and U2++, tested on test set in CER (%).*

training method	decoding mode	decoding chunk size	
		full	16
U2 Transformer	ctc prefix beam search	7.84	8.68
	attention rescoring	6.71	7.31
U2++ Transformer	ctc prefix beam search	7.45	8.21
	attention rescoring	6.09	6.70
U2 Conformer	ctc prefix beam search	7.02	7.76
	attention rescoring	6.08	6.46
U2++ Conformer	ctc prefix beam search	6.70	7.39
	attention rescoring	5.39	5.78

Table 4: *Ablation study of U2++, we remove the additional modules compared to the U2 model:(1) only removing R2L decoder; (2) only removing SpecSub ;(3) removing R2L decoder and SpecSub.*

model	full chunk	chunk 16
u2++ baseline	4.63	5.05
- R2L decoder	4.78	5.25
- SpecSub	4.78	5.22
- R2L decoder (U2)	4.97	5.45

Table 5: *Ablation study of the individual L2R decoder and R2L decoder using for re-scoring (RS) and auto-regressive beam search decoding method (AR) on AISHELL-1 test set. For U2: (1) reducing three decoder layers directly during training. For U2++: (1) only using R2L decoder during decoding; (2) only using L2R decoder during decoding. The results with same * or † denotes that it can be fair to directly compare.*

model	method	full chunk	chunk 16
U2 6layer *	AR	5.19	5.47
	RS	4.97	5.45
U2 3layers †	AR	5.27	5.55
	RS	4.99	5.49
U2++ baseline *	AR	4.83	5.12
	RS	4.63	5.05
only R2L decoder †	AR	5.07	5.35
	RS	4.81	5.19
only L2R decoder †	AR	5.01	5.30
	RS	4.76	5.15

4.3. Ablation Study

4.3.1. Proposed Improvement

We evaluate the effectiveness of the proposed SpecSub and the added R2L decoder loss. The results are shown in the Table 4.Both of them bring a significant improvement on the U2 baseline and the results are better when they are combined.

4.3.2. Individual Decoder Using in Decoding Process

We also evaluate the effectiveness of the two decoders in U2++ by using three different re-scoring methods, which are only the L2R score, only the R2L decoder score, and the combined score. The layer number of the U2 decoder is 6. To keep the parameter the same as U2++, we also train a U2 model with 3 decoder layers. In addition, we compare our re-scoring strategy with the auto-regressive strategy used in [14] in all settings. The details are reported in the Table 5.

The re-scoring decoding method still shows better results than the auto-regressive beam search in the U2++ model. No matter which decoding method is used, U2++ is better than U2 with three or six decoder layers. Even with only single decoders, the U2++ performs better than U2 with six decoder layers. When combining the scores of two direction decoders, U2++ achieves 7% relative CER reduction with U2.

The U2++ also shows the ability to make a trade-off between RTF/latency and CER. When lower RTF/latency is the top priority, users could use a single L2R decoder in exchange for better RTF.

4.3.3. Score Weights

The weight α is studied with different values during the training process include 0.1, 0.3, 0.5. We search the α and λ during the decoding process. The influence of α during the training process is detailed in Table 6. While α is set 0.3 during training, the decoding result is better than the rest.

Table 6: *Study on α value during the training process on AISHELL-1 test set, All the results are conduct by Conformer U2++ model.*

weight α	full chunk	chunk 16
0.1	4.74	5.09
0.3	4.63	5.05
0.5	4.67	5.13

Table 7: *Study on the number of R2L and L2R decoder layers on AISHELL-1 test set.*

decoder layers	full chunk	chunk 16
L2R=5, R2L=1	4.74	5.09
L2R=4, R2L=2	4.75	5.13
L2R=3, R2L=3	4.63	5.05

4.3.4. Allocation of Decoder Layer

In a fixed total parameters numbers setting, the parameters allocated to L2R and R2L decoders may influence the model performance. Our experiments on a total of six layers show that allocating the number of the parameters equally on the two decoders performs best. The details are reported in Table 7.

5. Conclusions

In this paper, we analyze the shortcoming of U2, which is the lack of modeling the right words in the attention decoder. So we propose the R2L decoder to involve the right information. We also proposed the SpecSub data augmentation method to make the model trained by dynamic chunk training more robust. Future works include exploring the decoder model consistent in the training and decoding process, such as a non-aggressive attention-based decoder. And more experiments are needed on the larger industrial datasets.

6. References

[1] 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*, 2006, pp. 369–376.

[2] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning*, 2016, pp. 173–182.

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

[4] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2013, pp. 6645–6649.

[5] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent nn: First results," in *NIPS 2014 Workshop on Deep Learning*, 2014.

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

[7] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in neural information processing systems*, 2015, pp. 577–585.

[8] B. Zhang, D. Wu, Z. Yao, X. Wang, F. Yu, C. Yang, L. Guo, Y. Hu, L. Xie, and X. Lei, "Unified streaming and non-streaming two-pass end-to-end model for speech recognition," *arXiv preprint arXiv:2012.05481*, 2020.

[9] B. Zhang, D. Wu, C. Yang, X. Chen, Z. Peng, X. Wang, Z. Yao, X. Wang, F. Yu, L. Xie *et al.*, "Wenet: Production first and production ready end-to-end speech recognition toolkit," *arXiv preprint arXiv:2102.01547*, 2021.

[10] X. Song, Z. Wu, Y. Huang, D. Su, and H. Meng, "Specswap: A simple data augmentation method for end-to-end speech recognition," *Proc. Interspeech 2020*, pp. 581–585, 2020.

[11] 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 *Proc. Interspeech*, 2019, pp. 2613–2617.

[12] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*. IEEE, 2017, pp. 1–5.

[13] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 4835–4839.

[14] X. Chen, S. Zhang, D. Song, P. Ouyang, and S. Yin, "Transformer with bidirectional decoder for speech recognition," *arXiv preprint arXiv:2008.04481*, 2020.

[15] Y. Higuchi, S. Watanabe, N. Chen, T. Ogawa, and T. Kobayashi, "Mask ctc: Non-autoregressive end-to-end asr with ctc and mask predict," *arXiv preprint arXiv:2005.08700*, 2020.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[17] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, 2020, pp. 5036–5040.

[18] Z. Tian, J. Yi, Y. Bai, J. Tao, S. Zhang, and Z. Wen, "Synchronous transformers for end-to-end speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7884–7888.

[19] 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 *Proc. Interspeech 2020*, 2020, pp. 2142–2146.

[20] H. Inaguma, M. Mimura, and T. Kawahara, "Enhancing monotonic multihead attention for streaming asr," in *Proc. Interspeech*, 2020, pp. 2137–2141.

[21] Z. Wang, W. Yang, P. Zhou, and W. Chen, "Wnars: Wfst based non-autoregressive streaming end-to-end speech recognition," *arXiv preprint arXiv:2104.03587*, 2021.