LFEformer: Local Feature Enhancement Using Sliding Window With Deformability for Automatic Speech Recognition

Guangyong Wei, Zhikui Duan, Shiren Li, Xinmei Yu, and Guangguang Yang

Abstract—**A module using sliding window with deformablity, abbreviated as SWD, has been proposed for local feature enhancement. In particular, the proposed SWD module adopts windows with variable size based on the depth of the embedded network layers. Moreover, the proposed SWD module is inserted into the Transformer network, referred as LFEformer, for automatic speech recognition. Such network is particularly good at capturing both local and global features, and this is beneficial for model improvement. It is worth mentioning that the local and global features are extracted by SWD module and the attention mechanism in Transformer network, respectively. The effectiveness of the LFEformer has been validated on three widely used datasets, which are Aishell-1, HKUST and WSJ (*dev93/eval92*). The experimental results demonstrate that 0.5% CER, 0.8% CER and 0.7%/0.3% WER improvement can be obtained in the correspondent datasets.**

Index Terms—**Speech Recognition, Transformer, Deformability, Local Feature.**

I. Introduction

THE recent proposed network named Transformer [1] has been extensively applied in various application scenarios, such as computer vision (CV) [2], [3], natural language processing (NLP) [4], [5] and automatic speech recognition (ASR) [6], [7], [8], [9]. When it comes to the ASR community, Transformer network has achieved good performance and become one of the most popular approaches. Thus, in the field of ASR, a great number of Transformer-based models have been developed to further improve the model performance and meet the need of speech recognition, such as local feature enhancement and feature fusion.

The key part in Transformer network is attention mechanism and this mechanism is used to establish long-range dependencies, which is regarded as global features instead of local features. In this way, the local features cannot be effectively obtained. To address this problem, some researchers focus on the scheme of local feature extraction and then the extracted local features are fed into the network for model improvement. [10] made use of convolution and attention mechanisms to capture the local and global context information. After that, some novel approaches, like [11], have been proposed to extract local attention under the help of a fixed-size sliding window and then add them into the global features. [12] found that the range of tokens concerned in each Transformer layer is different. For this reason, it is unreasonable to set the fixed sliding window when extracting local information and add them to other layers. Based on the above description, conclusions can be drawn that two issues hold the potential harm to the model performance. On the one hand, self-attention mechanism is inappropriate for local feature extraction. On the other hand, restricting the interaction among tokens with fixed window is a suboptimal solution for local feature extraction. To address these two problems, a module named sliding window with deformability (SWD) has been put forward to capture the robust local features. These features are further integrated into the global ones extracted by conventional Transformer for final prediction.

Features from higher level pay more attention to semantic information but the detail information are weak, while features from the shallow layers mainly focus on detail information. To a large extent, the detail information play a significant role in the prediction of the final result in ASR field. Thus, feature fusion is a desirable solution to this problem. For taking advantage of the detail information, the attention information from the previous layer is directly used as the prior knowledge of the current layer [13]. [14] selected a learnable and randomly initialized attention matrix as prior knowledge. Recently, [15] found that better performance can be obtained when features from embedding layer are embedded into other layers.

With the inspiration of the above researches, the proposed SWD module is used to extract local information from the embedding layer. In particular, the extracted features are fed into both encoder and decoder of every layer in the Transformer network. SWD is employed for local feature extraction via sliding windows with variable length. In this study, a novel network called LFEformer has been proposed with the integration between SWD module and the conventional Transformer network. The usefulness of the LFEformer network has been validated by extensive experiments on three widely-used datasets. The contributions are summarized as follows.

- The proposed LFEformer network can fully exploit both global features extracted by attention mechanism and local features extracted by SWD module for final prediction. The mentioned operation contributes to the model improvement in ASR domain.

Manuscript received 2 November 2022; revised 21 January 2023; accepted 27 January 2023. Date of publication 2 February 2023; date of current version 7 March 2023. This work was supported by Research Foundation under Grants 2020A1515111107, 2021B1515120025, and 2021KSYS008. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Rosangela Coelho. *(Corresponding author: Zhikui Duan.)*

Guangyong Wei, Zhikui Duan, Xinmei Yu, and Guangguang Yang are with the Foshan University, Foshan, Guangdong 528011, China (e-mail: gy.wei@foxmail.com; duanzhikui@outlook.com; labxmyu@fosu.edu.cn; 240991183@qq.com).

Shiren Li is with the Sun Yat-Sen University, Guangzhou 510275, China (e-mail: lishr6@mail3.sysu.edu.cn).

Digital Object Identifier 10.1109/LSP.2023.3241558

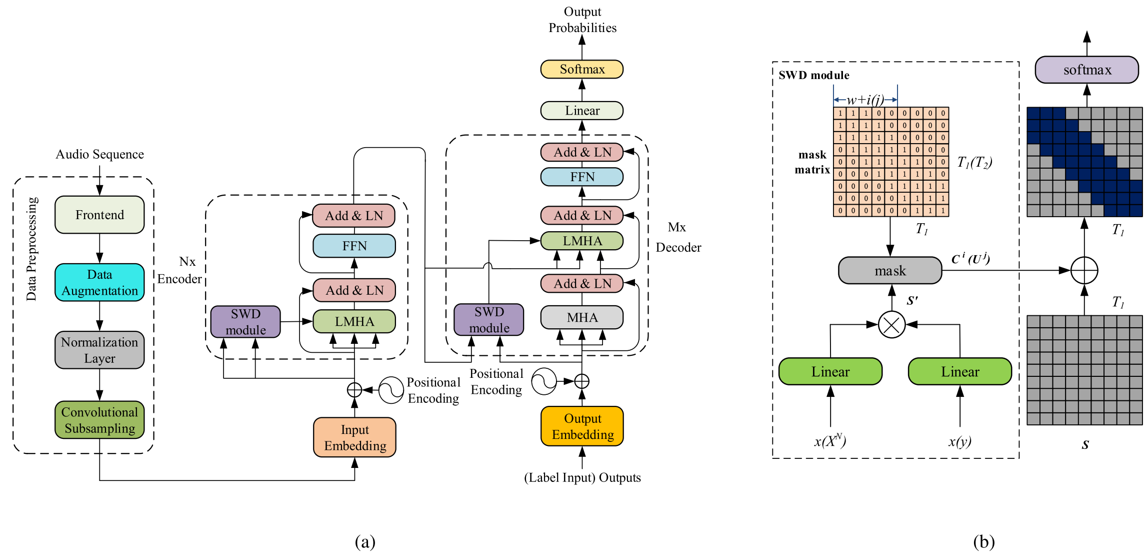

Fig. 1. (a) is the overall architecture of the proposed LFEformer network. The LFEformer consists of three parts: data preprocessing, encoder and decoder. Here *add* means matrix addition operation and *LN* means layer normalization. (b) is the structure of the SWD module. i represents the index of the current encoder layer and correspondingly j is the index of the current decoder layer. w represents the range of feature enhancements in the first layer of the encoder(decoder).

- The SWD module, as a local feature extractor, can effectively extract features from embedding layer. It should be highlighted that the extracted features are fed into each layer via sliding window with variable window size based on the depth of the layers. Moreover, the interaction among tokens from embedding layer is only calculated once, so the parameters in SWD module can be almost negligible compared with Transformer.
- A lot of experiments have been conducted to verify the effectiveness of the proposed LFEformer network. Compared with the original Transformer, the LFEformer can reduce 0.5% CER with only 0.32 M increase of model parameters on the Aishell-1 dataset. 0.8% CER improvement can be achieved on the HKUST dataset, and a 0.7%/0.3% WER improvement on the WSJ (*dev93*/*eval92*) can be achieved.

II. Methodology

In this section, the overall model structure and important details will be introduced first, following that the proposed SWD module is described in detail.

A. Overall Structure of LFEformer

As shown in Fig. 1(a), LFEformer consists of three parts: data preprocessing, encoder and decoder. Data preprocessing is needed prior to feeding into the network. The input and output of the data preprocessing model are raw audio sequences and log Mel-Filterbank, respectively. Spectrogram enhancement and downsampling are involved in this precedure. Each encoder layer contains a local enhanced multi-head self-attention (LMHA) module, a SWD module and a feed-forward network (FFN) module. Similar structure can be found in the decoder, but it has one more multi-head self-attention (MHA) module. Under the help of these modules, LFEformer can select different local attention ranges based on the layer depth, while retaining the ability to obtain global attention. Encoder and decoder are composed of N encoder layers and M decoder layers, respectively. Each module in the model contains residual connections [16] and layer normalization [17].

The LFEformer is a Transformer-based network, so their structures are similar. The major difference is that features extracted from SWD are fused into those from the attention mechanism. Thus, the details of the feature fusion are presented and the other content can refer to [1]. Given that X_1 and X_2 are the inputs of the SWD module and S is the attention score obtained from the attention mechanism, so the presentation of the integrated attention score can be expressed as :

$$attn = softmax\left[\frac{F(S'(X_1, X_2), mask(w, X_1, X_2)) + S}{\sqrt{d_{model}}}\right], \tag{1}$$

where $S'(\cdot)$ represents the interaction. $mask$ is the mask matrix generated for the correspondent layer, where w is the initial sliding window extent. $F(\cdot)$ is the dot product operation. The output of $F(\cdot)$ is combined with S to obtain the attention score. A scaling factor $\frac{1}{\sqrt{d_{model}}}$ is used to avoid the effect of too large attention scores on the $softmax$ function.

B. Local Feature Extraction From Embedding Layer

The attention mechanism in Transformer is good at extracting long-range context information, while the local features extraction capability is undesirable. According to past research findings [9], [18], [19], the local features are of great importance for model improvement in ASR domain. In addition, the input of current layer is the output of the previous layer and the features are passed throughout several layers. In this way, the features from the top layer contain little detail information. In a Transformer-based network, both the label sequences and the high-level acoustic features, which is viewed as the output of encoder, are important to the final prediction. The high-level acoustic features are fed into every decoder layer while the label sequences are only fed into the first layer. Therefore, the higher decoder layers often have weak label sequences information.

The second step is to obtain constrained local features. The SWD module uses sliding windows with different sizes to extract local features from the embedding layer. At last, the enhancement of local features is completed in the final stage together with the fusion between the local embedding layer information and traditional attention score of each layer.

As shown in Fig. 1(b), it is feasible for the SWD module to extract local features from the embedding layer. In the encoder, two inputs of the SWD module, viewed as the acoustic feature $\mathbf{x} = (x_1,..., x_{T_1})$ (T_1 is the length of the acoustic feature), are the same. The input $\mathbf{x}$ is transformed via two different linear layers, and then the dot product operation is performed to obtain the attention score of the embedding layer. The attention score limits the token interaction range using a mask matrix with a sliding window in size $w + i$. After that, the attention score C^i which is fused to the i-th layer can be obtained. In the decoder, the inputs of the SWD module are the output of the last encoder layer X^N and the label sequence $\mathbf{y} = (y_1,..., x_{T_2})$ (T_2 is the length of the label sequence). The attention score U^j fused to the j-th layer of the decoder can be obtained. The equations for C^i and U^j can be expressed as:

$$C^i = F[S'(x, x), mask(w + i, x, x)], \tag{2}$$

$$U^j = F[S'(X^N, y), mask(w + j, X^N, y)], \tag{3}$$

$$S'(X_1, X_2) = (W^q X_1 + b_1)(W^k X_2 + b_2)^T, \tag{4}$$

where $S'(\cdot)$, $mask$ and $F(\cdot)$ have similar meaning as in (1). In the mask matrix, the value is '1' in the range of the diagonal $w + i$, and '0' in other positions. In addition, $i \in [1, \ldots, N]$, $j \in [1, \ldots, M]$, $w \in \mathbb{N}$, $W^q \in \mathbb{R}^{d_{model} \times d_{model}}$ and $W^k \in \mathbb{R}^{d_{model} \times d_{model}}$ are weight matrices. b_1 and b_2 are bias vectors. $w + i$ represents the size of the sliding window in the SWD module at the i-th encoder layer. The outputs of the SWD module in encoder and decoder are $C^i \in \mathbb{R}^{T_1 \times T_1}$ and $U^j \in \mathbb{R}^{T_2 \times T_1}$, which are fused with the attention scores of the i-th encoder layer and the j-th decoder layer, respectively.

III. Experiment

A. Experimental Setting

Three widely-used datasets, Aishell-1 [20], HKUST [21] and WSJ [22], are selected to validate the effectiveness of the proposed LFEformer network. As shown in Fig. 1(a), the input audio signal acquires effective feature transformation at the frontend, including framing, windowing, Fast Fourier Transform (FFT) and Discrete Cosine Transform (DCT), and finally serves as the input of the encoder. Here, the window size is 25 ms. The window shift is 10 ms, and the output feature is 80 dimensional log Mel-filterbank.

All experiments are conducted on the ESPNet toolkit [23]. In particular, the label smoothing method and dropout regulation $p = 0.1$ have been selected to prevent the model from over fitting. Adam [24] is used as the optimizer, where the learning rate is 0.002, warmup steps is 25000, ϵ is 10^{-9}, $\beta 1$ is 0.9, and $\beta 2$ is 0.98. In addition, other parameters are set as follows: $d_{model} = 256$, $d_{ff} = 2048$, $w = 6$, $h = 4$, $N = 12$ and $M = 6$. With regard to the attention dimension: we adopt $d_q = d_k = d_v = d_{model}/h = 64$. The initial value of each batch is 64. $beam\_size$ in the beam search algorithm is 10. The language model (LM) is based on the Transformer framework, and the LM module has 16 layers and 15 epochs.

TABLE I
Comparison Among LFEformer and Other ASR Models in Aishell-1 Dataset

Model	Dev (%)	Test (%)
Transformer without LM [1]	5.5	5.9
Transformer with LM [1]	5.3	5.6
RNN-T [25]	10.13	11.82
SA-T [25]	8.30	9.30
Chunk-Flow SA-T [25]	8.58	9.80
Sync-Transformer [26]	7.91	8.91
Masked-NAT [6]	6.4	7.1
ESPNet-RNN [27]	6.8	8.0
Insertion-NAT [28]	6.1	6.7
LASO [29]	5.8	6.4
AT [30]	5.5	5.9
Realformer with LM [13]	5.7	6.1
LFEformer without LM	5.1	**5.4**
LFEformer with LM	5.0	**5.1**

TABLE II
Performance of ASR Models in HKUST Dataset

Model	CER (%)
Chain-TDNN [31]	23.7
Self-attention Aligner [32]	24.1
Extended-RNA [33]	26.6
Joint CTC-attention model/ESPNet [34]	27.4
SAM with LM [32]	24.92
CTC with LM [35]	34.8
Transformer with LM [1]	21.5
Transformer without LM [1]	21.7
Conformer without LM [8]	20.1
LFEformer with LM	**20.7**
LFEformer without LM	**20.9**
Conformer+SWD without LM	**19.8**

TABLE III
Comparison of WER Results Between LFEformer and Transformer Baseline in WSJ Dataset

Model	dev93 (%)	eval92 (%)
DeCoAR [36]	8.3	4.6
SAN-CTC[37]	8.9	5.9
E2E-SincNet [38]	7.8	4.7
Transformer with LM [1]	6.6	4.6
LFEformer with LM	**5.9**	**4.3**

B. Comparison With State-of-The-Art

From Table I, one can note that the proposed LFEformer in Aishell-1 has achieved better performance compared with other popular models. Compared with the baseline Transformer, one can note that the performance of LFEformer on the Dev dataset and testing dataset has correspondingly improved 0.3% CER and 0.5% CER when language model (LM) is involved, and 0.4% CER and 0.5% CER if LM is absent. The parameters of LFEformer is 30.68 M, which is only 0.32 M larger than the baseline Transformer.

Experimental results on the HKUST dataset are presented in Table II. From the table, one can find that 0.8% CER improvement can be achieved with or without LM compared with Transformer network. In order to further validate the the usefulness and the generality of proposed SWD module, the SWD module has been applied to Conformer network, which is denoted as "Conformer+SWD without LM" in Table II. Based on the results shown in Table II, one can note the 0.3% CER improvement can be gotten.

In addition, the proposed LFEformer is tested in English dataset named WSJ, except for the above two Chinese language datasets. The results are shown in Table III. Compared with the

TABLE IV
EMBEDDING LAYER INTERACTIONS ARE EXTRACTED IN DIFFERENT WAYS

Window Type	CER(%)	△
Transformer with LM	21.5	-
AEFI with LM	21.3	0.2% ↑
FSW with LM	21.1	0.4% ↑
SWD with LM	20.7	0.8% ↑

TABLE V
THE CHANGE TREND OF SWD IS COMPARED

Change Trend	CER (%)	△
Transformer with LM	21.5	-
IWS with LM	20.7	0.8% ↑
DWS with LM	20.9	0.6% ↑

baseline Transformer, *dev93* and *eval92* can achieve 0.7% and 0.3% WER improvements respectively.

Based on the experimental results, one can note that the proposed LFEformer network performs well in these three datasets and can surpass the state-of-the-art models.

C. Ablation Studies

The contribution of the proposed SWD module has been systematically investigated in this section. The HKUST dataset is selected for validation.

First, we experimentally verify the effect by using different approaches, such as fusing all embedding layer feature interactions (AEFI), fixed sliding windows (FSW) and SWD, to extract the interaction of the embedding layer. Specifically, when all elements in the mask matrix in (2) and (3) equal to 1, the SWD is identical with AEFI. When $i = j = 0$ in (2) and (3), the SWD is the same as FSD. Thus, AEFI and FSW are the special case of SWD. The results are reported in TABLE IV. From the table, one can note that the CER of AEFI, FSW and SWD are correspondingly improved by 0.2%, 0.4% and 0.8% compared with the baseline Transformer. Thus, it is logical to say that the interaction relationship of the embedding layer is beneficial for model improvement. Moreover, selecting different window ranges for different layers helps to further improve the model performance.

Two strategies, which are increasing window size (IWS) and decreasing window size (DWS), are adopted to explore the influence of changing window size on model performance. In particular, IWS is similar with that in SWD. When i in (2) and j in (3) are changed to $-i$ and $-j$, IWS is the same with DWS. The results are given in Table V. Compared with the baseline Transformer, the CER of IWS and DWS are improved by 0.8% and 0.6%, respectively. From the result, one can note that both strategies show positive effect to model performance, but IWS gives better outcome.

Finally, experiments are conducted to find which locations are appropriate to insert the SWD module, including the candidate locations like encoder self-attention (ESA), decoder self-attention (DSA) and decoder cross-attention (DCA). The results are given in Table VI. From the table, one can find that the proposed SWD module has no negative impact on the model performance regardless of the position, although the model performance does not change when it is only applied to the DSA. In addition, the best results can be obtained when SWD is inserted into ESA and DCA simultaneously. The model performance can be improved by 0.7% CER.

TABLE VI
THE APPLICATION OF THE SWD MODULE IN THE ENCODER AND DECODER ATTENTION MODULES

ESA	DSA	DCA	CER (%)
✓			20.8
	✓		21.3
		✓	21.5
✓	✓		20.8
✓		✓	**20.7**
	✓	✓	21.3
✓	✓	✓	20.8

Based on the above mentioned ablation studies, conclusions can be drawn that it is beneficial to fuse the interaction from the embedding layer with variable-length window to other layers. In addition, the IWS is good for model improvement.

IV. CONCLUSION

This paper proposes an ASR network called LFEformer, which consists of the original Transformer network and the SWD module. The SWD module, which acts as an independent branch, extracts the local features from the embedding layer via sliding window with different sizes based on the depth of the layer. Then the local features are fused into the global features extracted by attention mechanism in Transformer network. This operation enables the LFEformer to extract robust feature representations. Experimental results on three widely used datasets, HKUST, Aishell-1 and WSJ, have comprehensively validated the effectiveness of the proposed LFEformer. In particular, LFEformer can improve model performance with negligible additional model parameters.

REFERENCES

[1] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5999–6009.

[2] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.

[3] M. Zhao, G. Cao, X. Huang, and L. Yang, "Hybrid transformer-CNN for real image denoising," *IEEE Signal Process. Lett.*, vol. 29, pp. 1252–1256 2022.

[4] J. He and H. Hu, "Language reinforced superposition multimodal fusion for sentiment analysis," *IEEE Signal Process. Lett.*, vol. 29, pp. 1347–1351, 2022.

[5] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. T. X. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 2978–2988.

[6] N. Chen, S. Watanabe, J. Villalba, P. Żelasko, and N. Dehak, "Non-autoregressive transformer for speech recognition," *IEEE Signal Process. Lett.*, vol. 28, pp. 121–125, 2021.

[7] Z. Tian, J. Yi, J. Tao, S. Zhang, and Z. Wen, "Hybrid autoregressive and non-autoregressive transformer models for speech recognition," *IEEE Signal Process. Lett.*, vol. 29, pp. 762–766, 2022.

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

[9] Y. Peng, S. Dalmia, I. Lane, and S. Watanabe, "Branchformer: Parallel MLP-Attention architectures to capture local and global context for speech recognition and understanding," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 17627–17643.

[10] M. Burchi and V. Vielzeuf, "Efficient conformer: Progressive downsampling and grouped attention for automatic speech recognition," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2021, pp. 8–15.

[11] H. Zhang et al., "PoolingFormer: Long document modeling with pooling attention," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 12437–12446.

[12] K. Shim, J. Choi, and W. Sung, “Understanding the role of self attention for efficient speech recognition,” in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–19.

[13] R. He, A. Ravula, B. Kanagal, and J. Ainslie, “RealFormer: Transformer likes residual attention,” *ACL-IJCNLP*, 2021, pp. 929–943.

[14] Y. Tay, D. Bahri, D. Metzler, D. C. Juan, Z. Zhao, and C. Zheng, “Synthesizer: Rethinking self-attention for transformer models,” in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10183–10192.

[15] X. Liu et al., “Understanding and improving encoder layer fusion in sequence-to-sequence learning,” in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–14.

[16] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[17] J. L. Ba, J. R. Kiros, and G. Hinton, “Layer normalization,” 2016, *arXiv:1607.06450*.

[18] I. Beltagy, M. E. Peters, and A. Cohan, “Longformer: The long-document transformer,” 2020, *arXiv:2004.05150*.

[19] A. Joshua et al., “ETC: Encoding long and structured inputs in transformers,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 268–284.

[20] H. Bu, J. Du, X. Na, and H. Zheng, “AISHELL-1: An open-source mandarin speech corpus and a speech recognition baseline,” in *Proc. Conf. Oriental Chapter Int. Coordinating Committee Speech Databases Speech I/O Syst. Assessment*, 2017, pp. 1–5.

[21] Y. Liu, P. Fung, Y. Yang, C. Cieri, S. Huang, and D. Graff, “HKUST/MTS: A very large scale mandarin telephone speech corpus,” in *Proc. Int. Symp. Chin. Spoken Lang. Process.*, 2006, pp. 724–735.

[22] D. B. Paul and J. Baker, “The design for the wall street journal-based CSR corpus,” in *Proc. workshop Speech Natural Lang.*, 1992, pp. 357–362.

[23] S. Watanabe et al., “ESPnet: End-to-End speech processing toolkit,” in *Proc. Interspeech*, 2018, pp. 2207–2211.

[24] D. P. Kingma and J. Ba, “ADAM: A method for stochastic optimization,” in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–15.

[25] Z. Tian, J. Yi, J. Tao, Y. Bai, and Z. Wen, “Self-Attention Transducers for End-to-End Speech Recognition,” in *Proc. Interspeech*, 2019, pp. 4395–4399.

[26] Z. Tian, J. Yi, Y. Bai, J. Tao, S. Zhang, and Z. Wen, “Synchronous transformers for end-to-end speech recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 7884–7888.

[27] S. Karita et al., “A comparative study on transformer vs RNN in speech applications,” in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2019, pp. 449–456.

[28] Y. Fujita, S. Watanabe, M. Omachi, and X. Chan, “Insertion-based modeling for end-to-end automatic speech recognition,” *Interspeech*, 2020, pp. 3660–3664.

[29] Y. Bai, J. Yi, J. Tao, Z. Tian, Z. Wen, and S. Zhang, “Listen attentively, and spell once: Whole sentence generation via a non-autoregressive architecture for low-latency speech recognition,” in *Proc. Interspeech*, 2020, pp. 3381–3385.

[30] R. Fan, W. Chu, P. Chang, and J. Xiao, “CASS-NAT: CTC alignment-based single step non-autoregressive transformer for speech recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 5889–5893.

[31] D. Povey et al., “Purely sequence-trained neural networks for ASR based on lattice-free MMI,” in *Proc. Interspeech*, 2016, pp. 2751–2755.

[32] L. Dong, F. Wang, and B. Xu, “Self-attention Aligner: A latency-control end-to-end model for ASR using self-attention network and chunk-hopping,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 5656–5660.

[33] L. Dong, S. Zhou, W. Chen, and B. Xu, “Extending recurrent neural Aligner for streaming end-to-end speech recognition in mandarin,” in *Proc. Interspeech*, 2018, pp. 816–820.

[34] S. Kim, T. Hori, and S. Watanabe, “Joint CTC-attention based end-to-end speech recognition using multi-task learning,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 4835–4839.

[35] Y. Miao, M. Gowayyed, X. Na, T. Ko, F. Metze, and A. Waibel,, “An empirical exploration of CTC acoustic models,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2016, pp. 2623–2627.

[36] S. Ling, Y. Liu, J. Salazar, and K. Kirchhoff, “Deep contextualized acoustic representations for semi-supervised speech recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 6429–6433.

[37] J. Salazar, K. Kirchhoff, and Z. Huang, “Self-attention networks for connectionist temporal classification in speech recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 7115–7119.

[38] T. Parcollet, M. Morchid, and G. Linares, “E2E-SINCNET: Toward fully end-to-end speech recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 7714–7718.