NON-AUTOREGRESSIVE TRANSFORMER WITH UNIFIED BIDIRECTIONAL DECODER FOR AUTOMATIC SPEECH RECOGNITION

Chuan-Fei Zhang[1,2], *Yan Liu*[1,2,*], *Tian-Hao Zhang*[3], *Song-Lu Chen*[3], *Feng Chen*[3,4], *Xu-Cheng Yin*[3]

[1] School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China
[2] Key Laboratory of Knowledge Automation for Industrial Processes of Ministry of Education, Beijing 100083, China
[3] USTB-EEasyTech Joint Lab of Artificial Intelligence, University of Science and Technology Beijing, Beijing 100083, China
[4] EEasy Technology Company Ltd., Zhuhai 519000, China

ABSTRACT

Non-autoregressive (NAR) transformer models have been studied intensively in automatic speech recognition (ASR), and many NAR transformer models is to use the causal mask to limit token dependencies. However, the causal mask is designed for the left-to-right decoding process of the non-parallel autoregressive (AR) transformer, which is inappropriate for the parallel NAR transformer since it ignores the right-to-left contexts. Some methods are proposed to utilize right-to-left contexts with an extra decoder, but these methods increase the model complexity. To tackle the above problems, we propose a new non-autoregressive transformer with a unified bidirectional decoder (NAT-UBD), which can simultaneously utilize left-to-right and right-to-left contexts for ASR. However, direct use of bidirectional contexts will cause information leakage, which means the decoder output can be affected by the character information of the input in the same position. To avoid information leakage, we propose a novel attention mask and modify vanilla queries, keys, and values matrices for NAT-UBD. Experimental results verify that NAT-UBD can achieve character error rates (CERs) of 5.0%/5.5% on the Aishell-1 dev/test sets, outperforming all previous NAR transformer models. Moreover, NAT-UBD can run 49.8× faster than the AR transformer baseline when decoding in a single step.

Index Terms— automatic speech recognition, transformer, non-autoregressive, bidirectional contexts, information leakage.

1. INTRODUCTION

Recently, transformer models [1, 2] based on encoder-decoder have shown superior performance in end-to-end automatic speech recognition (ASR) compared with recurrent neural networks (RNNs) [3, 4] and connectionist temporal classification (CTC) [5]. Most transformer models predict the next token conditioning on encoded states and previously generated tokens in an autoregressive (AR) manner. However, the AR manner usually has slow decoding speed because of serial decoding.

To accelerate the decoding speed, non-autoregressive (NAR) transformer models [6, 7, 8] are first proposed in machine translation, which can predict multiple tokens simultaneously and have been widely studied in ASR recently. To our best knowledge, NAR transformer models in ASR can be roughly divided into two categories according to the decoder. The first kind of NAR transformer model [9, 10] regards the decoder as an acoustic model. However, such NAR transformer models adhere to the conditional independence hypothesis between the output tokens and suffer inferior recognition performance. The second kind of NAR transformer model [11, 12, 13, 14] regards the decoder as a language model, and the decoder can predict output conditioning on linguistic information. It is noteworthy that the attention mask is widely used in these NAR transformers to limit token dependencies. Especially, the causal mask proposed in the AR transformer [15] is used in the second kind of NAR transformers [13, 14] to construct a unidirectional decoder (Fig. 1 (a)). However, it is inappropriate for NAR transformer models to use the causal mask. Firstly, the causal mask is designed for the serial decoding process of the AR transformer while the decoding process of the NAR transformer is parallel. Secondly, the causal mask only utilizes left-to-right (L2R) contexts, resulting in discarded right-to-left (R2L) contexts.

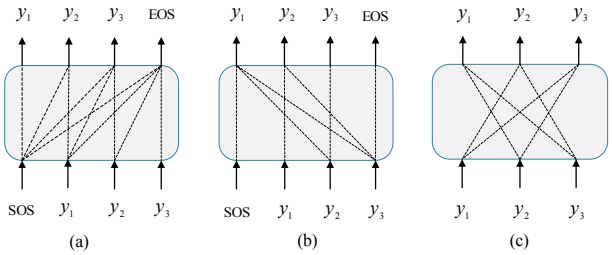

Fig. 1: (a) unidirectional decoder [13, 14] with left-to-right contexts only; (b) unidirectional decoder [16, 17] with right-to-left contexts only; (c) unified bidirectional decoder with left-to-right and right-to-left contexts simultaneously. The grey boxes denote transformer decoders, and the dashed lines denote token dependencies.

Previously, R2L contexts (Fig. 1 (b)) have been studied in the AR transformer [16] and the streaming ASR [17]. These models are composed of one shared encoder and two unidirectional decoders, i.e., two separate decoders with L2R and R2L contexts, respectively. Such a framework is complex and inefficient because it requires an extra unidirectional decoder and the two decoders have no information exchange.

To tackle the above problems, we propose a new non-autoregressive transformer with a unified bidirectional decoder (NAT-UBD), which can fully utilize both L2R and R2L contexts in a unified decoder (Fig. 1 (c)). However, direct use of bidirectional contexts will cause information leakage. Concretely, information leakage means the decoder output can be affected by the character information of the input in the same position, and the decoder can not refine the input during decoding. Since the proposed NAT-UBD is based on the speech transformer [1], the residual connection and self-attention are

* Corresponding author.

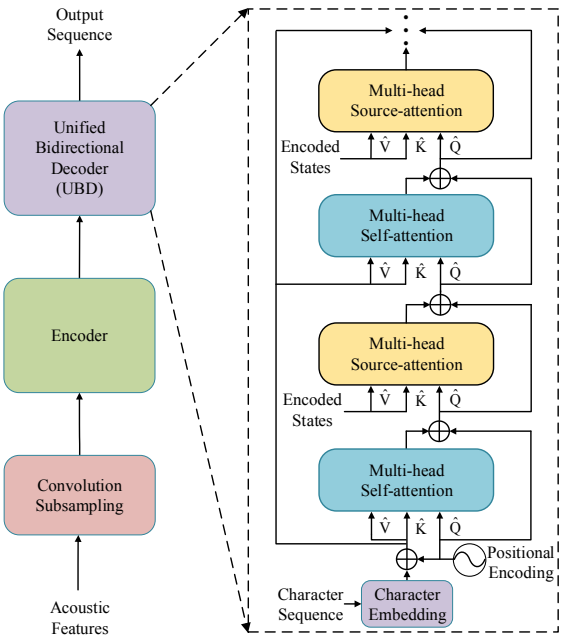

Fig. 2: The overall architecture of NAT-UBD. First, the input acoustic features are downsampled by convolutional subsampling layers and encoded by the encoder. Then the encoded states and character sequence are fed into UBD. Finally, UBD predicts all tokens simultaneously for accurate and fast recognition. The bidirectional character information is used in the multi-head self-attention layers.

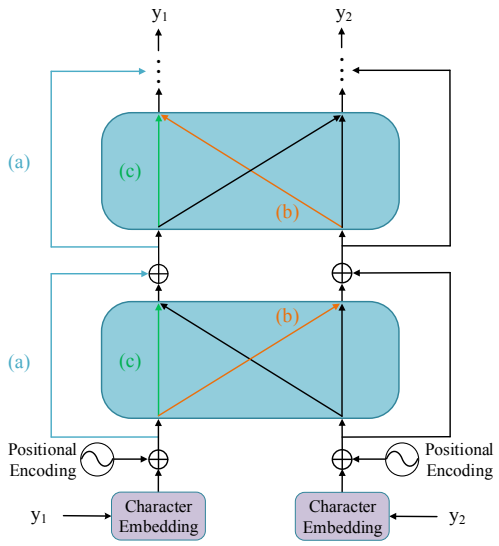

Fig. 3: Illustration of information leakage. Both residual connections and multi-head self-attention layers can cause information leakage. Arrows that can cause information leakage of y_1 are marked. (a) residual connections; (b) indirect attention connections; (c) direct attention connections. Blue boxes denote multi-head self-attention layers.

adopted, and both of them can cause information leakage. To avoid information leakage caused by the residual connection, we remove word embedding from the vanilla queries matrix (Q). To avoid information leakage caused by the self-attention, we propose a novel attention mask named self mask and make both keys matrix (K) and values matirx (V) independent of previous layers, similar to the Disco transformer [18]. This way, NAT-UBD can outperform all previous NAR transformer models on the Aishell-1 corpus and achieve competitive performance compared with the AR transformer baseline on the Magicdata corpus. Moreover, NAT-UBD can run 49.8× faster than the AR transformer baseline because all tokens can be predicted simultaneously.

2. METHODOLOGY

The proposed NAT-UBD can fully utilize L2R and R2L contexts in a unified decoder. Fig. 2 illustrates the overall architecture of NAT-UBD. For simplicity, we omit the feed-forward layers and layer normalization [19]. The convolutional subsampling layers and encoder of NAT-UBD are the same as the speech transformer [1], while the decoder is our proposed unified bidirectional decoder (UBD).

2.1. Unified Bidirectional Decoder

UBD takes the character sequence Y as the input and predicts all tokens simultaneously conditioning on bidirectional contexts $Y_{\neq t}$ and encoded states S, as described in Eq. (1).

$$y_t = UBD(Y_{\neq t}, S) \qquad 1 \leq t \leq T, \tag{1}$$

where T denotes the length of Y. $Y_{\neq t}$ represents that the output y_t should be prevented from utilizing the character information of the input in the same position. Otherwise, information leakage will arise, and UBD can not refine the input characters.

In the vanilla transformer, the character sequence Y is first transformed to $QKVs$ by the character embedding C and positional encoding P, as described in Eq. (2).

$$Q^1, K^1, V^1 = Linear(C(Y) + P) \tag{2}$$

For simplicity, we use $Linear()$ to represent different linear layers. This way, $QKVs$ of the first multi-head self-attention layer contain the character information, and the character information is transmitted by residual connections, position-wise feed-forward layers and multi-head self-attention layers. However, both residual connections (Fig. 3 (a)) and multi-head self-attention layers (Fig. 3 (b)(c)) can cause information leakage.

To avoid information leakage caused by residual connections (Fig. 3 (a)), all queries of Q should not contain the character information in the current position. So we remove the character embedding and calculate the queries matrix of the first multi-head self-attention layer as in Eq. (3).

$$\hat{Q}^1 = Linear(P), \tag{3}$$

where $\hat{Q}$ represents the queries matrix of NAT-UBD. However, KVs can not be modified like $\hat{Q}$ since the character information must be retained and utilized in UBD. An alternative method is to remove residual connections for KVs. Nevertheless, removing residual connections is not enough since the attention connections in multi-head self-attention layers can also cause information leakage.

To avoid the information leakage caused by indirect attention connections (Fig. 3 (b)), all keys and values in multi-head self-attention layers should only contain the character information in

the current position. So we feed the same keys and values matrices independent of previous layers into all the multi-head self-attention layers, as described in Eq. (4).

$$\hat{K}^i, \hat{V}^i = Linear(C(Y) + P) \quad 1 \leq i \leq I, \tag{4}$$

where $\hat{K}$ and $\hat{V}$ represent keys and values matrices of NAT-UBD. Besides, I is the number of multi-head self-attention layers.

To avoid the information leakage caused by direct attention connections (Fig. 3 (c)), all positions in the attention weight matrix should mask out attention connections of themselves. Hence, we propose an attention mask named self mask to conduct the Hadamard product with the attention weight matrix. Concretely, the self mask is a matrix of which diagonal elements are set to 0, and all the other elements are set to 1. After conducting the Hadamard product, the diagonal elements of the attention weight matrix are set to 0. Meanwhile, the self mask can make UBD utilize bidirectional contexts simultaneously.

2.2. Joint Training

We use the label sequence as the input character sequence of UBD. However, the label sequence is not available during decoding. We thus additionally apply the CTC loss [5] to the encoder to enable the encoder to output preliminary results for UBD. Then NAT-UBD can be jointly trained with both the CTC loss L_{CTC} and UBD loss L_{UBD}, as described in Eq. (5).

$$L = \lambda L_{CTC} + (1 - \lambda) L_{UBD}, \tag{5}$$

where λ is used to balance two losses. L_{UBD} represents the cross entropy loss [20].

2.3. Decoding by Iterative Refinement and Adaptive Termination

To achieve fast decoding speed, we substitute the label sequence with the greedy CTC output as the initial UBD input. Every decoder input can be predicted with bidirectional contexts, and the whole output sequence can be predicted via iterative refinement in J iterations, as described in Eq. (6).

$$\hat{y}_t^j = \begin{cases} Decoder(\hat{Y}_{\neq t}^{j-1,c}, S_e) & j = 1, \\ Decoder(\hat{Y}_{\neq t}^{j-1,d}, S_e) & 1 < j \leq J, \end{cases} \tag{6}$$

where $\hat{Y}^{j-1,c}$ is the greedy CTC output and $\hat{Y}^{j-1,d}$ is the greedy decoder output.

Moreover, we propose a simple and effective stop method named adaptive termination for NAT-UBD to accelerate decoding speed. When the output of the jth iteration is the same as the $(j-1)$th iteration, the iteration can be early terminated because the output of subsequent iteration will remain unchanged.

3. EXPERIMENTS

3.1. Datasets

The experiments are carried out on the 178-hour Aishell-1 Mandarin corpus [21] and 755-hour Magicdata Mandarin corpus[1]. For the input acoustic features, we extract 80-channel filterbanks features, computed from a 25ms window with a stride of 10ms. The output labels consist of 4231 Chinese characters for Aishell-1 and 4518 Chinese characters for Magicdata, obtained from the training set.

Table 1: Character error rate (CER) and real time factor (RTF) on the Aishell-1 corpus. † means SpecAugment is used, and ‡ means Speed Perturbation is used. J is the maximum number of iterations.

Model	J	Dev	Test	RTF	Params(M)
AR transformer					
Transformer [22]‡	–	6.0	6.7	–	29.7
NAR transformer					
LASO-big [12]†‡	1	5.8	6.4	–	≈ 105.0
KERMIT [28]‡	1	6.7	7.5	–	–
InDIGO [28]‡	1	6.0	6.7	–	–
CASS-NAT [10]†‡	1	5.3	5.8	–	29.7
CTC-enhanced [13]†‡	1	5.3	5.9	–	29.7
A-FMLM [29]‡	1	6.2	6.7	–	–
TSNAT-small [14]‡	1	5.4	5.9	–	34.0
ST-NAR [9]‡	1	6.88	7.67	–	≈ 31.0
Ours					
AR transformer†‡	–	5.24	5.57	0.4034	29.7
NAT-UBD†‡	1	5.13	5.62	**0.0081**	29.7
NAT-UBD†‡	10	**5.02**	**5.49**	0.0116	29.7

3.2. Experimental Setup

We conduct all experiments on ESPNet [22], which is an end-to-end speech processing toolkit. The convolutional subsampling module is comprised of 2 CNN layers with size 3×3, filter 256, stride 2 on the time dimension for 4× down-sampling. Then 12 encoder layers and 6 decoder layers are stacked. We use 256 dimensions for $\hat{Q}\hat{K}\hat{V}s$ and 4 attention heads for all multi-head attention layers. We set 2048 dimensions and use the ReLU activation function for position-wise feed-forward layers. Label smoothing [23] with a penalty of 0.1 is applied to prevent over-fitting. We use Adam [24] optimizer and warm up with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$. All models are trained on 2 Titan X GPUs with batch size 32. Gradients are accumulated [25] over 4 iterations. SpecAugment [26] is used for data augmentation, and Speed Perturbation [27] is additionally used for the Aishell-1 corpus. We use the dev set for early stopping. We choose the models with the top-10 lowest accuracies on the dev set and average them to get the final model. All decoding processes are performed utterance by utterance on a Titan X GPU without any external language model. Character error rate (CER) and real time factor (RTF)[2] are adopted for model evaluation. RTF is computed as the ratio of the total decoding time to the total duration of the test set.

3.3. Comparative Experiments

Except for NAT-UBD, we reimplement the AR transformer baseline with CTC joint training and decoding [30]. Especially, beam search with a width of 10 is used for the AR transformer baseline. As shown in Table 1, our proposed NAT-UBD achieves the best CERs than all previous NAR transformer models with different iterations and can outperform the AR transformer baseline with a 49.8× faster decoding speed on the Aishell-1 corpus.

[1] https://www.magicdatatech.cn/datasets

[2] Decoding speed can be affected by the hardware and utterance length. For example, the decoding speed of the AR model usually is in approximate linear relation to utterance length.

Table 2: Character error rate (CER) and real time factor (RTF) on the Magicdata corpus. † means SpecAugment is used.

Model	J	Dev	Test	RTF	Params(M)
Ours					
AR transformer†	–	**4.14**	**4.62**	0.4703	29.7
NAT-UBD†	1	4.39	4.85	**0.0095**	29.7
NAT-UBD†	10	4.23	4.70	0.0121	29.7

Table 3: Ablation study of $\hat{Q}\hat{K}\hat{V}s$ and self mask on the Aishell-1 corpus. $J = 0$ means that the greedy CTC output is directly used as the final output.

Model	J	Dev	Test
NAT-UBD	0	5.48	6.04
	10	**5.02**	**5.49**
$-\hat{Q}\hat{K}\hat{V}s$	0	12.67	13.62
	10	12.67	13.62
−self mask	0	11.13	12.61
	10	11.13	12.61
−both	0	10.58	11.33
	10	10.58	11.33

We also conduct experiments on the Magicdata Mandarin corpus. Magicdata is a large ASR corpus, and to our best knowledge, we are the first to evaluate the NAR transformer models on this corpus. From Table 2,We can see that NAT-UBD can achieve competitive CERs with the AR transformer baseline while maintaining faster decoding speed, proving the generalizability of NAT-UBD.

3.4. Necessity of Avoiding Information Leakage

To verify the necessity of avoiding information leakage, we replace $\hat{Q}\hat{K}\hat{V}s$ and self mask with vanilla $QKVs$ and padding mask, respectively. The padding mask is an attention mask that only masks the attention connections of padded tokens in a mini-batch and is widely used in transformer models [11, 12, 13, 14].

After replacing $\hat{Q}\hat{K}\hat{V}s$ or self mask, the accuracies on the dev set reach saturation quickly, and the training processes are stopped because of early stopping, resulting in few training epochs and poor CTC performance. However, we only focus on the difference between the greedy CTC output and decoder output. As shown in Table 3, except for NAT-UBD, the decoder output of the other three models is the same as the greedy CTC output, indicating that these three decoders are stuck in identity mapping between input and output during training. The experimental phenomena verify that information leakage can damage the network performance during decoding, proving both the $\hat{Q}\hat{K}\hat{V}s$ and self mask should be used.

3.5. Effectiveness of Adaptive Termination

We remove the adaptive termination from NAT-UBD during decoding. From Table 4, we can conclude that the adaptive termination can improve the decoding speed more than $2\times$ times with $J = 10$ while keeping CERs unchanged.

3.6. Effectiveness of Unified Bidirectional Decoder

We verify the effectiveness of UBD by replacing UBD with the L2R decoder and R2L decoder. To ensure the total parameters unchanged, when using L2R and R2L decoders simultaneously, each decoder only has 3 decoder layers compared with using 6 decoder layers for other methods. Besides, adaptive termination is used for all methods during decoding.

Table 4: Ablation study of adaptive termination on the Aishell-1 corpus.

Model	J	Dev	Test	RTF
NAT-UBD	10	5.02	5.49	**0.0116**
−adaptive termination	10	5.02	5.49	0.0263

Table 5: Comparison of the left-to-right (L2R) decoder, right-to-left (R2L) decoder, and unified bidirectional decoder (UBD) on the Aishell-1 corpus and Magicdata corpus.

Decoder	J	Aishell-1			Magicdata		
		Dev	Test	RTF	Dev	Test	RTF
L2R	1	5.37	5.91	0.0081	4.72	5.04	0.0095
	10	5.46	5.90	0.0108	4.66	5.01	0.0115
R2L	1	5.39	5.92	0.0081	4.74	4.99	0.0095
	10	5.55	6.17	0.0103	4.71	5.08	0.0110
L2R+R2L	1	5.64	6.18	**0.0077**	4.84	5.22	**0.0090**
	10	5.62	6.29	0.0091	4.78	5.21	0.0101
UBD	1	5.13	5.62	0.0081	4.39	4.85	0.0095
	10	**5.02**	**5.49**	0.0116	**4.23**	**4.70**	0.0121

As shown in Table 5, UBD achieves the best CERs on two corpora with $J = 1$. With $J = 10$, the CERs of UBD can be further decreased on two corpora while the other methods decrease a little or get even worse. In cases of using L2R and R2L decoders simultaneously, the decoding speed is faster than the other methods due to the parallel decoding of two decoders on a GPU. However, this method yields worse CERs than only using a unidirectional decoder. It is because the two decoders have no information exchange, which means their output only depend on unidirectional contexts. Moreover, both decoders only have 3 decoder layers, which can affect the feature extraction ability compared with using 6 decoder layers. As a result, we can conclude that the proposed UBD is efficient and can use ample linguistic information for character prediction.

However, the RTF of UBD is higher than the other methods on both Aishell-1 and Magicdata corpora with $J = 10$. When using UBD, we observe that the cyclic dependency of adjacent tokens often arises, which can invalidate the adaptive termination and reduce the decoding speed. For example, the label sequence is "stand up", and the output of the first iteration is "stand down". Then the output of the second and third iterations might be "sit up" and "stand down". As a result, the iteration process can not stop until 10 iterations. Especially, cyclic dependency arises with increasing frequency when the phonetic pronunciation is similar.

4. CONCLUSION

We propose a new non-autoregressive transformer with a unified bidirectional decoder (NAT-UBD), carefully designed to simultaneously utilize left-to-right and right-to-left contexts and prevent consequent information leakage. As a result, the proposed NAT-UBD outperforms all previous NAR transformer models on the Aishell-1 corpus and achieves competitive performance with the AR transformer baseline on the Magicdata corpus without any external language model. For the decoding speed, NAT-UBD can run $49.8\times$ faster than the AR transformer baseline. Extensive experiments prove the effectiveness of the unified bidirectional decoder and the necessity of avoiding information leakage.

5. REFERENCES

[1] Linhao Dong, Shuang Xu, and Bo Xu, "Speech-Transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *ICASSP*, 2018, pp. 5884–5888.

[2] Yingzhu Zhao, Chongjia Ni, Cheung-Chi Leung, Joty Shafiq, Eng Siong Chng, and Bin Ma, "Speech transformer with speaker aware persistent memory," in *INTERSPEECH*, 2020, pp. 1261–1265.

[3] Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, et al., "State-of-the-art speech recognition with sequence-to-sequence models," in *ICASSP*, 2018, pp. 4774–4778.

[4] Bo Li, Shuo-Yiin Chang, Tara N. Sainath, Ruoming Pang, Yanzhang He, et al., "Towards fast and accurate streaming end-to-end asr," in *ICASSP*, 2020, pp. 6069–6073.

[5] Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006, pp. 369–376.

[6] Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher, "Non-autoregressive neural machine translation," in *ICLR*, 2018.

[7] Marjan Ghazvininejad, Vladimir Karpukhin, Luke Zettlemoyer, and Omer Levy, "Aligned cross entropy for non-autoregressive machine translation," in *ICML*, 2020, pp. 3515–3523.

[8] Jason Lee, Elman Mansimov, and Kyunghyun Cho, "Deterministic non-autoregressive neural sequence modeling by iterative refinement," in *EMNLP*, 2018, pp. 1173–1182.

[9] Zhengkun Tian, Jiangyan Yi, Jianhua Tao, Ye Bai, Shuai Zhang, et al., "Spike-triggered non-autoregressive transformer for end-to-end speech recognition," in *INTERSPEECH*, 2020, pp. 5026–5030.

[10] Ruchao Fan, Wei Chu, Peng Chang, and Jing Xiao, "CASS-NAT: CTC alignment-based single step non-autoregressive transformer for speech recognition," in *ICASSP*, 2021, pp. 5889–5893.

[11] Yosuke Higuchi, Shinji Watanabe, Chen Nanxin, Tetsuji Ogawa, et al., "Mask CTC: non-autoregressive end-to-end asr with CTC and Mask predict," in *INTERSPEECH*, 2020, pp. 3655–3659.

[12] Ye Bai, Jiangyan Yi, Jianhua Tao, Zhengkun Tian, Zhengqi Wen, et al., "Listen attentively, and spell once: Whole sentence generation via a non-autoregressive architecture for low-latency speech recognition," in *INTERSPEECH*, 2020, pp. 3381–3385.

[13] Xingchen Song, Zhiyong Wu, Yiheng Huang, Chao Weng, Dan Su, et al., "Non-autoregressive transformer asr with CTC-Enhanced decoder input," in *ICASSP*, 2021, pp. 5894–5898.

[14] Zhengkun Tian, Jiangyan Yi, Jianhua Tao, Ye Bai, Shuai Zhang, Zhengqi Wen, and Xuefei Liu, "TSNAT: two-step non-autoregressvie transformer models for speech recognition," *arXiv preprint arXiv:2104.01522*, 2021.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, et al., "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.

[16] Xi Chen, Songyang Zhang, Dandan Song, Peng Ouyang, and Shouyi Yin, "Transformer with bidirectional decoder for speech recognition," in *INTERSPEECH*, 2020, pp. 1773–1777.

[17] Di Wu, Binbin Zhang, Chao Yang, Zhendong Peng, Wenjing Xia, et al., "U2++: Unified two-pass bidirectional end-to-end model for speech recognition," *arXiv preprint arXiv:2106.05642*, 2021.

[18] Jungo Kasai, James Cross, Marjan Ghazvininejad, and Jiatao Gu, "Non-autoregressive machine translation with disentangled contexts transformer," in *ICML*, 2020, pp. 5144–5155.

[19] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton, "layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[20] Pieter-Tjerk de Boer, Dirk P.Kroese, Shie Mannor, and Reuven Y.Rubinstein, "A tutorial on the cross-entropy method," *Annals of operations research*, vol. 134, no. 1, pp. 19–67, 2005.

[21] Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng, "Aishell-1: An open-source mandarin speech corpus and aspeech recognition baseline," in *O-COCOSDA*, 2017, pp. 1–5.

[22] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, et al., "Espnet: end-to-end speech processing toolkit," in *ICASSP*, 2018, pp. 2207–2211.

[23] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna, "Rethinking the Inception architecture for computer vision," in *CVPR*, 2016, pp. 2818–2826.

[24] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[25] Myle Ott, Sergey Edunov, David Grangier, and Michael Auli, "Scaling neural machine translation," in *WMT*, 2018, pp. 1–9.

[26] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, et al., "Specaugment: A simple data augmentation method for automatic speech recognition," in *INTERSPEECH*, 2019, pp. 2613–2617.

[27] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "Audio augmentation for speech recognition," in *INTERSPEECH*, 2015, pp. 3586–3589.

[28] Yuya Fujita, Shinji Watanabe, Motoi Omachi, and Xuankai Chang, "Insertion-based modeling for end-to-end automatic speech recognition," in *INTERSPEECH*, 2020, pp. 3660–3664.

[29] Nanxin Chen, Shinji Watanabe, Jesús Villalba, and Najim Dehak, "Listen and fill in the missing letters: Non-autoregressive transformer for speech recognition," *arXiv preprint arXiv:1911.04908*, 2020.

[30] Shigeki Karita, Nelson Enrique Yalta Soplin, Shinji Watanabe, Marc Delcroix, Atsunori Ogawa, and Tomohiro Nakatani, "Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration," in *INTERSPEECH*, 2020, pp. 1408–1412.