Hybrid Autoregressive and Non-Autoregressive Transformer Models for Speech Recognition

Zhengkun Tian, *Student Member, IEEE*, Jiangyan Yi, *Member, IEEE*, Jianhua Tao, *Senior Member, IEEE*, Shuai Zhang, *Student Member, IEEE*, and Zhengqi Wen, *Member, IEEE*

***Abstract*—The autoregressive (AR) models, such as attention-based encoder-decoder models and RNN-Transducer, have achieved great success in speech recognition. They predict the output sequence conditioned on the previous tokens and acoustic encoded states, which is inefficient on GPUs. The non-autoregressive (NAR) models can get rid of the temporal dependency between the output tokens and predict the entire output tokens in one inference step. However, the NAR model still faces two major problems. Firstly, there is still a great gap in performance between the NAR models and the advanced AR models. Secondly, it's difficult for most of the NAR models to train and converge. We propose a hybrid autoregressive and non-autoregressive transformer (HANAT) model, which integrates AR and NAR models deeply by sharing parameters. We assume that the AR model will assist the NAR model to learn some linguistic dependencies and accelerate the convergence. Furthermore, the two-stage hybrid inference is applied to improve the model performance. All the experiments are conducted on a mandarin dataset ASIEHLL-1 and a english dataset librispeech-960 h. The results show that the HANAT can achieve a competitive performance with the AR model and outperform many complicated NAR models. Besides, the RTF is only 1/5 of the AR model.**

***Index Terms*—Autoregressive, non-autoregressive, transformer, hybrid, speech recognition.**

I. Introduction

THE AR models, especially attention-based encoder-decoder models [1]–[4] and transducer-based models [5]–[9], have achieved success in speech recognition. Most of them generate the target sequence in an autoregressive fashion, which predicts the next token conditioned on the previously generated tokens and the acoustic encoded sequence. The autoregressive characteristic makes the inference process be carried out step by step, which cannot be implemented in parallel and results in a large latency.

By contrast, the NAR model can get rid of the temporal dependency and directly generate the target sequences based on the encoded acoustic states in at least one inference step. Although the NAR models can perform inference very efficiently, it still faces two significant problems. Firstly, there is still a great gap in performance between the advanced AR model and the NAR model. Chen et.al proposed two kinds of iterative inference methods to alleviate the problem [10]. Three inference iterations have a negative impact on the speed of inference. Besides, some researchers also utilized a CTC Model to generate the preliminary predictions and then correct the previous prediction by a non-autoregressive decoder [11]–[13]. However, it is hard to eliminate accumulated errors caused by the CTC Models and carry out the frame-wise operation of the CTC module in parallel. Secondly, it is difficult for most of the NAR transformer models to train and converge. To our knowledge, there are three major ways to alleviate this problem. Firstly, some works try to introduce an auxiliary CTC loss to accelerate the training and convergence [12], [14]. Secondly, more iterations in the training process also bring the improvement on the performance [10], [15]. Thirdly, instead of learning from a sequence partially or fully filled with <MASK> [10], [15], some works try to improve training efficiency by providing a preliminary sequence with the information of target sequence to the non-autoregressive transformer decoder [11]–[14]. These methods are either time-consuming or difficult to implement. We notice that there are two main similarities between the transformer-based AR and NAR models. Firstly, they have similar model structure and parameters [10]. A typical AR transformer model can be directly converted into a NAR transformer model by removing the mask-operation of masked self-attention in the decoder. Secondly, both of them predict the target sequence appended with the end-of-sentence tokens. Similar output patterns indicate that the probability distribution they learn may be very close. Based on these two observations, we guess that it is feasible to combine the AR and NAR models deeply. In this paper, we propose a model named the hybrid autoregressive and non-autoregressive transformer (HANAT) model, which consists of a front-end convolutional block, an acoustic encoder, and a dual-mode transformer decoder. The dual-mode transformer decoder can model the context in both an autoregressive and a non-autoregressive way. Different from the traditional method that transfers knowledge into the NAR model from a pre-trained AR model [20], we train an AR model and a NAR model from scratch simultaneously. The NAR model can learn some linguistic prior knowledge from

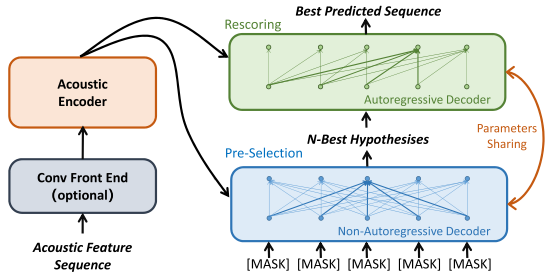
Manuscript received July 30, 2021; revised February 3, 2022; accepted February 11, 2022. Date of publication February 16, 2022; date of current version March 22, 2022. This work was supported in part by the National Key Research and Development Plan of China under Grant 2020AAA0140003, in part by the National Natural Science Foundation of China (NSFC) under Grants 61901473 and 61831022, in part by the Key Research Project under Grant 2019KD0AD01, in part by Inria-CAS Joint Research Project under Grant 173211KYSB20190049, and in part by Huawei Noah's Ark Lab. The associate editor coordinating the review of this manuscript and approving it for publication was Mr. Ville M. Hautamaki *(Corresponding authors: Jianhua Tao; Jiangyan Yi.)*

Zhengkun Tian and Shuai Zhang are with the NLPR Lab, CASIA, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100190, China (e-mail: zhengkun.tian@nlpr.ia.ac.cn; shuai.zhang@nlpr.ia.ac.cn).

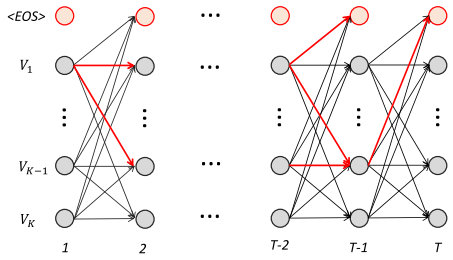
Jiangyan Yi and Zhengqi Wen are with the NLPR Lab, CASIA, Beijing 100190, China (e-mail: jiangyan.yi@nlpr.ia.ac.cn; zqwen@nlpr.ia.ac.cn).

Jianhua Tao is with the NLPR Lab, CASIA, Beijing 100190, China, and also with the CAS Center for Excellence in Brain Science and Intelligence Technology, Beijing 100190, China (e-mail: jhtao@nlpr.ia.ac.cn).

Digital Object Identifier 10.1109/LSP.2022.3152128



(a) The Model Structure and Inference Process



(b) The Inference Graph Under NAR Mode

Fig. 1. (a) illustrates the structure of our proposed hybrid autoregressive and non-autoregressive transformer (HANAT) and the inference process. The HANAT model consists of an optional front-end block, an acoustic encoder, and a dual-mode transformer decoder. The inference process can be divided into two steps, pre-selection and rescoring. The dual-mode transformer decoder first generates N-best hypothesizes under the NAR mode and then rescores these hypotheses in an autoregressive way. (b) illustrates an output probability graph of the non-autoregressive model. The first red circle of each column means the end-of-sentence token <EOS>. The other black circle represents different tokens of vocabulary. Each path that starts with the blank circle and ends with a red circle in the graph represents a possible hypothesis.

the AR model depending on the dual-mode transformer decoder. Furthermore, we introduce the two-pass method [17]–[19] into the inference process of our model, which improves the model performance greatly. All the experiments are conducted on a Chinese mandarin dataset ASIEHLL-1 and a english dataset librispeech-960. The results show that the HANAT can achieve a competitive performance with the AR models and outperform other NAR models. Furthermore, compared with the AR model with the same parameters, the RTF of our HANAT is reduced nearly five times.

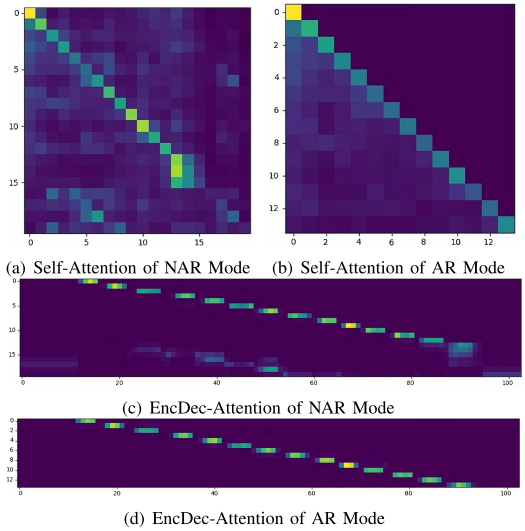
II. Our Proposed Model

Our proposed hybrid autoregressive and non-autoregressive transformer, as shown in Fig. 1(a), consists of three components, a front-end convolutional block, an acoustic encoder, and a dual-mode transformer decoder.

A. The Convolutional Front End and Acoustic Encoder

The front-end convolutional block generally consists of two 2D-Convolution layers and an output project layer [4]. The convolution layer is mainly responsible for the processing and down-sampling of the low-level acoustic features. The last output project layer will project the processed acoustic feature into the dimension that the encoder required.

We utilize the transformer encoder [4], [22] as the acoustic encoder, which contains a positional embedding, N repetitive multi-head self-attention layers (MHA), and feed-forward network layer (FFN). The sine-cosine positional embedding proposed by [22] is applied for all the experiments in this paper.



(a) Self-Attention of NAR Mode (b) Self-Attention of AR Mode

(c) EncDec-Attention of NAR Mode

(d) EncDec-Attention of AR Mode

Fig. 2. The Visualization of Attention Weights in The Dual-Mode Decoder. We extract the self-attention weights and encoder-decoder attention weights from the last block of the dual-mode transformer decoder. The pictures come from the test set, and its corresponding sentence ID is 'BAC009S0764W0121'.

Besides, the model also applies residual connection and layer normalization. The feed-forward network layer (FFN) contains two linear layers and a gated liner unit (GLU) [23] activation function [7].

B. The Dual-Mode Transformer Decoder

The major difference between the autoregressive model and the non-autoregressive model focus on the structure of the decoder. The AR models force themselves to learn the linguistic dependencies by blocking out the future information. This characteristic makes the AR model predict the output sequence step by step. The conditional probability $P_{AR}(Y|X)$ can be expressed as:

$$P_{AR}(Y|X) = P(y_1|X)\prod_{i=2}^{L} P(y_i|y_{<i}, X) \tag{1}$$

where X is the acoustic encoded sequence with length T and y_i denotes the i-th token of predicted sequence with length L. However, the non-autoregressive models get rid of the dependencies on the previously predicted tokens. Each step in the inference process is independent of each other, which makes the inference step can be implemented in parallel and greatly improves the reasoning speed of the model. The conditional probability $P_{NAR}(Y|X)$ can be rewritten as:

$$P_{NAR}(Y|X) = \prod_{i=1}^{L} P(y_i|X) \tag{2}$$

Both the AR decoder and the NAR decoder can apply the transformer as the basic structure. There are two significant differences between these two kinds of decoders. Firstly, the AR transformer needs to model the linguistic dependencies by applying the masking operation to the previous output, but the NAR transformer discards the masking operations, which makes it the output tokens conditional independent. Secondly, the AR decoder adopts all the tokens of vocabulary and some special tokens as the modeling units, while the NAR decoder

only requires one <MASK> token. Their modeling units are not coincident.

Inspired by the dual-mode ASR [16], which models the streaming ASR and non-streaming ASR by sharing an encoder, we propose a novel method that unifies the AR and NAR decoder into one dual-mode transformer decoder (DMTD). Obviously, there are some similarities in model structure and output patterns between these two models. We assume that the AR decoder will accelerate the training and convergence of the NAR decoder. The linguistic dependencies learned by the AR decoder might be able to provide some prior knowledge for the NAR decoder, which will make the training of the NAR decoder more efficient than guessing based on the empty sequence. The self-attention of DMTD applies to the masking-optional operation. During training, the encoder forwards once, and the decoder forwards twice, once in autoregressive mode and once in non-autoregressive mode. During the AR forward, the DMTD will adopt the truth token sequences that begin with the begin-of-sentence token <BOS> as the input, and then mask future information of self-attention and calculate the cross-entropy (CE) loss $\mathcal{L}_{AR}$. For the NAR forward, the decoder will utilize a sequence filled with <MASK> as the input and calculate the CE loss $\mathcal{L}_{NAR}$. The final loss is equal to the weighted sum of $\mathcal{L}_{AR}$ and $\mathcal{L}_{NAR}$.

$$\mathcal{L} = (1-\alpha)\mathcal{L}_{NAR} + \alpha\mathcal{L}_{AR} \tag{3}$$

where α dictates the weight of AR loss. The model is optimized by these two losses jointly.

C. The Two Step Hybrid Inference

The AR model predicts the output tokens conditioned on the previously generated tokens and the acoustic encoded states, which makes the model have high accuracy but low inference speed. Most of the non-autoregressive models generate the probability matrix in one step and then just select the token which has the highest probability at each position as the final predicted result [10], [14], [15]. This improves the inference speed, but cannot model the dependencies between the output tokens. Inspired by the success of two-pass models [17]–[19], we integrate the AR and NAR inference deeply. Now that the dual-mode transformer decoder is able to be trained in the AR and NAR fashion simultaneously, the decoder can perform dual-mode inference naturally.

The inference process can be divided into two steps, pre-selection and rescoring. During the first pre-selection, the dual-mode decoder will generate the conditional probability matrix from a full-mask sequence in a NAR way and then select the N-best hypothesizes. As shown in Fig. 1(b), we consider the conditional probability matrix generated by the decoder as a graph, which contains numerous possible hypothesize. Each hypothesis starts with any blank circle in the first column and ends with a red circle (end-of-sentence token <EOS>). In order to avoid that the model tends to choose shorter sentences, we utilize the length-normalized scores. The selection of N-best hypothesizes from the probability graph contains only simple addition operations, and does not take much time and computing resources. During the second step, the dual-mode decoder will insert begin-of-sentence token <BOS> into the head of the N-best hypothesizes. Then it utilizes the processed candidates as the input and predicts them in an AR fashion. Due to the transformer can carry out efficient computing in parallel, the entire inference process can be finished in two steps.

III. Experiments and Results

A. Dataset

In this work, all experiments are conducted on a public Mandarin speech corpus AISHELL-1[1] and a english corpus Librispeech-960 h.[2]

B. Experimental Setup

For all experiments, we use the 80-dimensional FBank features for AISHELL-1 and 83-dim combined FBank-Pitch feature for Librispeech, which are all computed on a 25 ms window with a 10 ms shift. For AISHELL-1, we choose 4234 characters (including a padding symbol <PAD>, an unknown token <UNK>, a begin-of-sentence token <BOS>) and an end-of-sentence token <EOS> as modeling units. In addition, we adopts 5004 modeling units (including 5000 wordpiece units generated by sentencepiece tookit) for librispeech.

Our model consists of 12 encoder blocks and 6 decoder blocks. There are 4 heads in multi-head attention. The 2D convolution front end utilizes two-layer time-axis CNN with ReLU activation, stride size 2, channels 384, and kernel size 3. Both the output size of the multi-head attention and the feed-forward layers are 384. The hidden size of the feed-forward layers is 768. We adopt an Adam optimizer with warmup steps 12000 and the learning rate scheduler reported in [22]. We train the model for 100 epochs on AISHELL-1 and 200 epochs on Librispeech-960 h respectively. And the model parameters of the last 20 epochs are averaged for the inference. For AISHELL-1, we adpots SpecAugment [24] (only including the time-masking and frequency-masking) for data augmentation. Both the SpecAugment and the 3-way speed perturbation are applied for librispeech-960 h.

We use the character error rate (CER) or word error rate (WER) to evaluate the performance of different models. For evaluating the inference speed of different models, we decode utterances one by one to compute real-time factor (RTF) on the test set. The RTF is the time taken to decode one second of speech. All experiments are conducted on a GeForce GTX TITAN X 12 G GPU.

C. Results

1) Comparison of the Model With Different AR Weights α: This section compares the models with the different AR weights α. When the weight α is equal to 0, the NAR model is equivalent to the one that adopts an empty sequence as the input without any prior knowledge. If α is equal to 1, the model will be completely transformed into an AR model. When the weight is between 0 and 1, the model can perform the two-step inference. For all one-step non-autoregressive inference in this section (marked as OneStep), the NAR models ($\alpha \in [0, 1)$) apply greedy search and the AR ($\alpha = 1.0$) models apply the beam search with width 10. The two-step inference of this section (marked as TwoStep) rescores the 10-best candidate sequences. As shown in Table I, it's obvious that the hybrid AR and NAR training ($\alpha \in (0, 1)$) can improve the performance of the NAR model. We guess that the NAR decoder can get some prior knowledge from the AR decoder by sharing the parameters, which reduces the difficulty

[1][Online]. Available: https://openslr.org/33/

[2][Online]. Available: https://www.openslr.org/12/

TABLE I
COMPARISON OF THE MODEL WITH DIFFERENT AR WEIGHTS α (CER %)

Weight α		0.0	0.3	0.5	0.7	0.9	1.0
Dev	OneStep	6.5	5.9	**5.8**	**5.8**	6.1	5.3
	TwoStep	-	5.6	5.5	**5.4**	5.6	-
Test	OneStep	7.2	6.5	6.5	**6.4**	6.5	6.0
	TwoStep	-	6.2	6.1	**6.0**	6.2	-

TABLE II
COMPARISON OF THE INFLUENCE OF THE N-BEST SEQUENCES ON THE MODEL PERFORMANCE (CER %)

N	1	5	10	20	50
Dev	5.8	5.4	5.4	5.4	**5.3**
Test	6.4	6.0	**5.9**	**5.9**	**5.9**
RTF	0.0054	0.0167	0.0173	0.0181	0.0220

TABLE III
THE PARAMETER CONFIGURATION OF THE MODEL

Mode	HANAT-Small	HANAT-Middle	HANAT-Big
d_{model}	384	512	512
d_{ff}	768	1024	2048
#Params	34M	59M	87M

of training the NAR model from scratch. In the case of one-step inference, there is still a great performance gap between the AR model ($\alpha = 1.0$) and the NAR model ($\alpha \in [0, 1)$). However, the introduced two-step inference method can improve the performance of the NAR model greatly and achieve comparable results with the AR model.

2) The Influence of the N-Best Sequences on the Model Performance: This section pays attention to the inference of the N-best sequence on the NAR model performance. We select the different number of candidate sequences. Under the condition that N is equal to 1, the two-step inference makes no sense. Therefore, we replace it with the result of the greedy search. The results in Table II indicate that the more candidate sequences, the better performance it archives, which is consistent with the results previously reported [19]. When the number of candidate sequences is greater than 10, the performance tends to be stable. However, with the increase of N, the model requires more computing resources, which leads to an increase in latency and real-time-factor (RTF). Taken together, we will select the 10-best candidates for the second-step rescoring by balancing the model performance and the inference speed.

3) Comparison With Other Models: To further study the upper limit of model performance, we readjust the parameter configuration of the model and named three models of different sizes, named `HANAT-Small`, `HANAT-Middle` and `HANAT-Big`. The three models have the same depth (12 encoder blocks and 6 decoder blocks) and are trained under the same conditions. Their differences focus on the dimensions of the model (d_{model}) and the hidden size of the feed-forward network (d_{ff}), as shown in Table III. We conduct the experiments on AISHELL-1 and Libripseech-960. And the results are recorded in Tables IV and V.

Compared with the other non-autoregressive models, we proposed HANAT model can achieve quite competitive performance. The results also show that the two-step hybrid inference is faster than ST-NAT with a neural language model because the two-step inference of the dual-mode transformer decoder can be implemented in parallel without iteration step by step. What's more, the performance of HANAT is very close to the AR transformer model, but the inference speed is nearly five times faster than it.

TABLE IV
COMPARISON WITH OTHER MODELS (CER %). ALL MODELS IN THIS TABLE USE SPECAUGMENT TO IMPROVE THE PERFORMANCE

Model	LM	Dev	Test	RTF
A-FMLM(K=1) [10]	w/o	6.2	6.7	-
Insertion-NAT [25]	w/o	6.1	6.7	-
LASO-big [15] ◊	w/o	5.8	6.4	-
CASS-NAT [26] ◊	w	5.3	5.8	-
CTC-enhanced NAR [13] ◊	w/o	5.3	5.9	-
ST-NAT [14]	w/o	6.9	7.7	0.0056
ST-NAT [14]	w	6.4	7.0	0.0292
AR-Transformer-Small (34M)	w/o	5.3	5.9	0.0557
AR-Transformer-Middle (59M)	w/o	5.2	5.7	0.0613
AR-Transformer-Big (87M)	w/o	**5.0**	5.6	0.0721
HANAT-Small (34M) †	w/o	5.8	6.4	0.0054
Two Step Hybrid Inference	w/o	5.4	5.9	0.0173
HANAT-Middle (59M) †	w/o	5.4	6.0	0.0063
Two Step Hybrid Inference	w/o	5.2	5.7	0.0176
HANAT-Big (87M) †	w/o	5.3	6.0	0.0077
AR Inference	w/o	5.2	5.7	0.0735
Two Step Hybrid Inference	w/o	5.1	**5.6**	0.0185

◊ These models additionally use speed-perturb to augment the speech data.
† The default inference for HANAT is the one-step NAR decoding.

TABLE V
THE RESULT ON LIBRISPEECH-960 (WER %)

Model	Dev		Test	
	clean	other	clean	other
AR-Transformer-Small (34M)	5.0	13.2	5.1	13.9
HANAT-Small (34M)	8.1	18.8	8.3	19.5
Two Step Hybrid Inference	5.1	13.3	5.4	14.2

4) The Analysis of Attention Weights in the Dual-Mode Decoder: We want to find out the difference between the AR and NAR modes of the dual-mode transformer decoder. Therefore, we extract the attention weights of these two modes from the last decoder block, as shown in Fig. 2. It's clear that the self-attention mechanism of the NAR model focuses on the whole input sequence, while the self-attention mechanism of the AR mode pays attention to the previous sequence. This is consistent with their own characteristics. Besides, the encoder-decoder attention weights of the NAR mode are more dispersed than the one under the AR mode, which also shows the autoregressive model has a stronger modeling ability.

IV. CONCLUSION

Compared with the AR model the NAR model can get rid of the temporal dependency and predict the entire tokens in one step. However, the NAR model still faces two problems. Firstly, there is still a great performance gap between the advanced AR model and the NAR transformer model. Secondly, it's difficult for most of the NAR models to train and converge. In this paper, we propose a hybrid autoregressive and non-autoregressive model (HANAT), which integrates the AR and NAR model deeply. Our model can not only perform inference quickly but also remove the great performance gap between the AR and NAR models. The experiments show that our proposed model can achieve comparable performance with the AR model and outperform other typical NAR models.

REFERENCES

[1] C. J. K., B. Dzmitry, S. Dmitriy, C. Kyunghyun, and B. Yoshua, "Attention-based models for speech recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 577–585.

[2] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2016, pp. 4960–4964.

[3] K. Suyoun, H. Takaaki, and W. Shinji, "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.

[4] D. Linhao, X. Shuang, and X. Bo, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 5884–5888.

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

[6] Y. He *et al.*, "Streaming end-to-end speech recognition for mobile devices," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 6381–6385.

[7] 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.

[8] Q. Zhang *et al.*, "Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-T loss," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 7829–7833.

[9] C.-F. Yeh *et al.*, "Transformer-transducer: End-to-end speech recognition with self-attention," 2019, *arXiv:1910.12977*.

[10] N. Chen, S. Watanabe, J. Villalba, and N. Dehak, "Non-autoregressive transformer automatic speech recognition," *IEEE Signal Process. Lett.*, vol. 28, 2021, pp. 121–125, doi: 10.1109/LSP.2020.3044547.

[11] Y. Higuchi, S. Watanabe, N. Chen, T. Ogawa, and T. Kobayashi, "Mask CTC: Non-autoregressive end-to-end ASR with CTC and mask predict," in *Proc. Interspeech*, 2020, pp. 3655–3659, doi: 10.21437/Interspeech.2020-2404.

[12] Y. Higuchi, H. Inaguma, S. Watanabe, T. Ogawa, and T. Kobayashi, "Improved mask-CTC for non-autoregressive end-to-end ASR," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2021, pp. 8363–8367.

[13] X. Song, Z. Wu, Y. Huang, C. Weng, D. Su, and H. Meng, "Non-autoregressive transformer ASR with CTC-enhanced decoder input," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2021, pp. 5894–5898.

[14] Z. Tian, J. Yi, J. Tao, Y. Bai, S. Zhang, and Z. Wen, "Spike-triggered non-autoregressive transformer for end-to-end speech recognition," in *Proc. Interspeech*, 2020, pp. 5026–5030.

[15] 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.

[16] J. Yu, W. Han, A. Gulati, C.-C. Chiu, B. Li, T. N. Sainath, Y. Wu, and R. Pang, "Dual-mode ASR: Unify and improve streaming ASR with full-context modeling," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–13.

[17] T. N. Sainath *et al.*, "Two-pass end-to-end speech recognition," in *Proc. Interspeech*, 2019, pp. 2773–2777.

[18] K. Hu, T. N. Sainath, R. Pang, and R. Prabhavalkar, "Deliberation model based two-pass end-to-end speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 7799–7803.

[19] Z. Tian, J. Yi, Y. Bai, J. Tao, S. Zhang, and Z. Wen, "One in a hundred: Selecting the best predicted sequence from numerous candidates for speech recognition," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, 2021, pp. 454–459.

[20] Y. Ren *et al.*, "Fastspeech: Fast, robust and controllable text to speech," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 3171–3180.

[21] Y. Zhao, J. Li, X. Wang, and Y. Li, "The speechtransformer for large-scale Mandarin Chinese speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 7095–7099.

[22] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[23] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 933–941.

[24] D. S. Park *et al.*, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019, pp. 2613–2617.

[25] Y. Fujita, S. Watanabe, M. Omachi, and X. Chang, "Insertion-based modeling for end-to-end automatic speech recognition," in *Proc. Interspeech*, 2020, pp. 3660–3664.

[26] 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.