INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Multi-Speaker ASR Combining Non-Autoregressive Conformer CTC and Conditional Speaker Chain

Pengcheng Guo[1], *Xuankai Chang*[2], *Shinji Watanabe*[2], *Lei Xie*[1*]

[1]ASLP@NPU, School of Computer Science, Northwestern Polytechnical University, Xi'an, China
[2]Carnegie Mellon University, USA

pcguo@nwpu-aslp.org, xuankaic@andrew.cmu.edu, shinjiw@ieee.org, lxie@nwpu.edu.cn

Abstract

Non-autoregressive (NAR) models have achieved a large inference computation reduction and comparable results with autoregressive (AR) models on various sequence to sequence tasks. However, there has been limited research aiming to explore the NAR approaches on sequence to multi-sequence problems, like multi-speaker automatic speech recognition (ASR). In this study, we extend our proposed conditional chain model to NAR multi-speaker ASR. Specifically, the output of each speaker is inferred one-by-one using both the input mixture speech and previously-estimated conditional speaker features. In each step, a NAR connectionist temporal classification (CTC) encoder is used to perform parallel computation. With this design, the total inference steps will be restricted to the number of mixed speakers. Besides, we also adopt the Conformer and incorporate an intermediate CTC loss to improve the performance. Experiments on WSJ0-Mix and LibriMix corpora show that our model outperforms other NAR models with only a slight increase of latency, achieving WERs of 22.3% and 24.9%, respectively. Moreover, by including the data of variable numbers of speakers, our model can even better than the PIT-Conformer AR model with only 1/7 latency, obtaining WERs of 19.9% and 34.3% on WSJ0-2mix and WSJ0-3mix sets. All of our codes are publicly available at https://github.com/pengchengguo/espnet/tree/conditional-multispk.

Index Terms: Non-autoregressive, conditional chain model, multi-speaker speech recognition

1. Introduction

End-to-end architectures have demonstrated their effectiveness and became the dominant models across various sequence to sequence tasks, like neural machine translation (NMT) [1, 2] and automatic speech recognition (ASR) [3, 4, 5, 6, 7]. However, most of these models follow an autoregressive (AR) strategy, which predicts a target token conditioned on both previously generated tokens and the source input sequence. The incremental process makes it hard to compute parallel and results in a large latency during the inference. In contrary to AR models, non-autoregressive (NAR) models have drawn immense interest recently, aiming to get rid of the temporal dependency and perform parallel inference.

NAR models were first proposed in NMT and have achieved competitive performance with conventional AR models [8, 9, 10, 11, 12, 13, 14, 15]. The idea of NAR models is to predict the whole target sequence within a *constant* number of iterations which is not strict with the sequence length. In [8], Gu *et al.* introduced a fertility module to predict the number of times each encoder output should be repeated and regraded the repeated encoder outputs as decoder input to perform parallel inference. In [10], Lee *et al.* proposed a deterministic NAR model by iteratively refine the outputs from corrupted predictions. In addition, there were lots of studies based on the insert or edit sequence generation [11, 12], connectionist temporal classification (CTC) [9], and masked language model objective [13, 14, 15].

Inspired by the success of NAR models in NMT, several NAR methods were also proposed to reach the performance of AR models on ASR [16, 17, 18, 19, 20, 21, 22]. Since CTC learns a frame-wise latent alignment between the input speech and output tokens and predicts the target sequence based on a strong conditional independence assumption [23], it can be viewed as an early-stage realization of NAR ASR models. In [18], Imputer was proposed to iteratively generate a new CTC alignment based on mask prediction. Besides, Mask-CTC [17, 20] and Align-Refine [21] aimed to refine a token-level CTC output or latent alignments with the mask prediction. In [19], Tian *et al.* proposed to use the estimated CTC spikes to predict the length of target sequence and adopt the encoder states as the input of decoder. However, most of aforementioned methods mainly focus on sequence to sequence tasks, like NMT and single-speaker ASR, and it is hard to directly extended to sequence to multi-sequence tasks, like multi-speaker ASR.

Multi-speaker ASR aims to predict the corresponding transcription for each speaker from multiple speakers overlapping speech. Although lots of AR models were explored for multi-speaker ASR, such as permutation invariant training (PIT) [24] or deep clustering (DPCL) [25] based hybrid system and recurrent neural network (RNN) or Transformer based end-to-end model models [26, 27, 28, 29, 30], few attempts have been made to realize NAR training. In this study, we revisit the proposed conditional chain based methods [31, 32, 33, 34] and extend it to NAR multi-speaker ASR. By doing this, the output of each speaker is predicted one-by-one by making use of both the mixed input as well as previously-estimated conditional speaker features. In each prediction step, a CTC-based NAR encoder network is used to perform parallel computation. Since the performance of CTC may suffer a severe degradation due to the conditional independence assumption, we also explore adopting an advanced Conformer encoder [6] architecture to capture both local and global acoustic dependencies and an additional intermediate loss [35] as a regularization function. Finally, while the original conditional chain model takes the token-level CTC alignments as the "hard" conditional speaker features, we propose to use "soft" conditions which are latent feature representations extracted after the last encoder layer. We evaluate the effectiveness of our model on two multi-speaker ASR benchmarks, WSJ0-Mix and LibriMix. Both results outperform other NAR models with a minor increment of latency

* corresponding author

http://dx.doi.org/10.21437/Interspeech.2021-2155

and even achieve comparable results with the AR models.

2. End-to-end Multi-speaker ASR

We briefly introduce the end-to-end multi-speaker ASR in this section. Given the input features of the mixture speech $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$, where T means the number of frames, the target is to directly predict the transcriptions $\mathbf{Y} = \{\mathbf{Y}^1, \dots, \mathbf{Y}^J$ for different speakers, where J refers to the number of mixed speakers. In [26, 28], an end-to-end multi-speaker ASR model was proposed, which is based on the joint CTC/attention-based encoder-decoder framework [36]. The encoder of the model consists of a mixture encoder, speaker-dependent (SD) encoders, and a recognition encoder. For a mixture speech X, the encoder output can be formulated as:

$$\mathbf{H} = \text{Encoder}_{\text{Mix}}(\mathbf{X}), \tag{1}$$

$$\mathbf{H}^j = \text{Encoder}^j_{\text{SD}}(\mathbf{H}),\ j = 1, \dots, J, \tag{2}$$

$$\mathbf{G}^j = \text{Encoder}_{\text{Rec}}(\mathbf{H}^j),\ j = 1, \dots, J. \tag{3}$$

Firstly, the mixture encoder in Eq. (1) encodes the input features as hidden representations $\mathbf{H}$ of the mixture speech. Then, J speaker-different encoders extract each speaker's speech representation $\mathbf{H}^j$ from the mixture representation. Next, the recognition encoder maps speech representations of each speaker into the high-level embeddings $\mathbf{G}^j$. Following the encoder, a CTC objective function is used to train the encoder and determine the best permutation $\hat{\pi}$ of the embeddings by permutation invariant training (PIT) [37]:

$$\hat{\pi} = \arg\min_{\pi \in \mathcal{P}} \sum_j \text{Loss}_{\text{CTC}}(\mathbf{G}^j, \mathbf{Y}^{\pi(j)}),\ j = 1, \dots, J, \tag{4}$$

where $\mathbf{Y}^j$ is the j-th reference transcription and $\mathcal{P}$ means the set of all permutations on $\{1, \dots, J\}$.

After encoder, an attention-based decoder takes each high-level embedding $\mathbf{G}^j$ and generates the hypothesis $\widetilde{\mathbf{Y}}^j$. The computation of the decoder is:

$$\mathbf{c}^j_n = \text{Attention}(\mathbf{e}^j_{n-1}, \mathbf{G}^j), \tag{5}$$

$$\mathbf{e}^j_n = \text{Update}(\mathbf{e}^j_{n-1}, \mathbf{c}^j_{n-1}, \widetilde{y}^j_{n-1}), \tag{6}$$

$$\widetilde{y}^j_n \sim \text{Decoder}(\mathbf{e}^j_n, \widetilde{y}^j_{n-1}), \tag{7}$$

in which $\mathbf{c}^j_n$ denotes the context vector and $\mathbf{e}^j_n$ is the hidden state of the decoder at step n. The permutation computed in the CTC step (as in Eq. (4)) also plays an important role in the decoder, determining the order of the reference sequences for the cross-entropy (CE) loss function and the input history of teacher-forcing training. The final loss function is a combination of the CTC loss and the decoder CE loss:

$$\begin{aligned}\mathcal{L} = \sum_j \Big(&\lambda \text{Loss}_{\text{CTC}}(\mathbf{G}^j, \mathbf{Y}^{\hat{\pi}(j)}) + \\ &(1-\lambda)\text{Loss}_{\text{Attn}}(\widetilde{\mathbf{Y}}^j, \mathbf{Y}^{\hat{\pi}(j)})\Big),\end{aligned} \tag{8}$$

where λ is an interpolation factor to scale different losses.

3. Non-autoregressive Multi-speaker ASR

End-to-end models proposed in Section 2 mainly focus on an AR strategy, which will be cumbered with a complex computation and large latency problems. Although an encoder-only CTC framework can be regarded as a NAR model, the system may be susceptible to performance degradation due to the conditional independence assumption. In this study, we revisit our proposed conditional speaker chain based method for NAR multi-speaker ASR. The improved model consists of a conditional speaker chain module and Conformer CTC encoders. While the conditional speaker chain explicitly models the relevance between outputs of different iterations, the Conformer CTC aims to conduct NAR computation in each single step. The total inference steps are restricted to the number of mixed speakers. In addition, we also explore incorporating an intermediate CTC loss as a regularization function to further improve the system performance.

3.1. Conformer Encoder

Conformer encoder [6] is a stacked multi-block architecture, which includes a multihead self-attention (MHSA) module, a convolution (CONV) module, and a pair of positionwise feed-forward (FNN) module in the Macaron-Net style. While the MHSA learns the global context, the CONV module efficiently captures the local correlations synchronously. Since the Conformer encoder has shown consistent improvement over a wide range of end-to-end speech processing applications [7], we expect it to compensate for the modeling capacity of CTC and improve the system performance.

3.2. Intermediate CTC Loss

In [35], Lee *et al.* proposed a simple but efficient auxiliary loss function for CTC based ASR models, named intermediate CTC loss. The main idea of intermediate CTC loss is to choose an intermediate layer within the encoder network and induce a sub-network by skipping all higher layers after the selected layer. By computing the additional CTC loss w.r.t the output of intermediate layer, the sub-network relies more on the lower layers instead of the higher layers, which can regularize the model training. Choosing the m-th layer from a L-layer encoder network, its output can be defined as $\mathbf{H}^j_m$. Thus, the final loss of our model becomes:

$$\begin{aligned}\mathcal{L} = \sum_j \Big((1-\lambda)\text{Loss}_{\text{CTC}}(\mathbf{G}^j, \mathbf{Y}^{\pi(j)}) + \\ \lambda \text{Loss}_{\text{InterCTC}}(\mathbf{H}^j_m, \mathbf{Y}^{\pi(j)})\Big),\end{aligned} \tag{9}$$

where λ refers to the weight of intermediate loss. In this work, we set λ equals to 0.1 and choose a middle layer of the $\text{Encoder}_{\text{Rec}}$ as the intermediate layer ($m = L/2$).

3.3. Conditional Chain Model

Fig. 1 shows an overview of our model. Different from the AR models described in Section 2, we replace the SD encoders with a conditional speaker chain module (CondChain) and predict the output of each speaker one-by-one. With a hidden mixture representation $\mathbf{H}$ computed in Eq. (1), the CondChain module extracts each speaker's speech representation by taking advantage of both the mixture representation $\mathbf{H}$ and the previously-estimated high-level embedding $\mathbf{G}^{j-1}$:

$$\mathbf{H}^j = \text{CondChain}(\mathbf{H}, \text{Embed}(\mathbf{G}^{j-1})),\ j = 1, \dots, J, \tag{10}$$

where $\mathbf{G}^{j-1}$, obtaining from the $\text{Encoder}_{\text{Rec}}$ output for previous speaker, can be viewed as the speaker condition. The Embed module is a multi-layer fully connect layer aiming to project

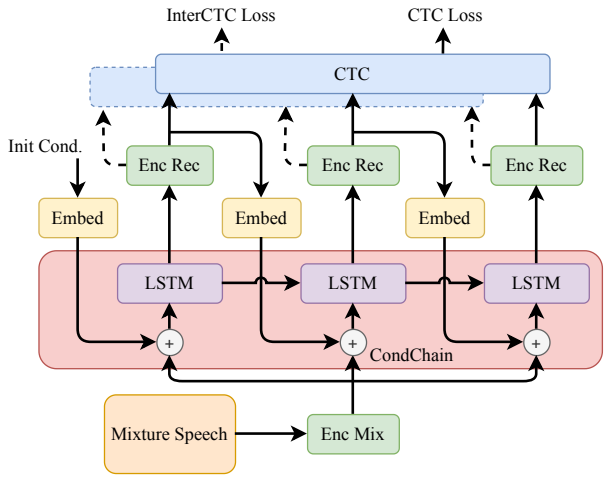

Figure 1: *A overview of proposed conditional speaker chain based Conformer CTC model for NAR multi-speaker ASR. This figure shows a training procedure of a 3-speaker mixed waveform. The parameters of blocks with the same name are shared.*

the linguistic sequence $\mathbf{G}^{j-1}$ into the acoustic sub-space. In the first step, an all-zero vector will be initialized as the speaker condition. Besides, the long short-term memory (LSTM) layer also helps to provide all historic speaker conditions by the flowing states. With this design, we can successfully perform a NAR computation in each step and the total inference steps is a constant number equaling to the number of mixed speakers. Moreover, compared with other multi-speaker ASR methods, which have to fix the number of mixed speakers in the training data, our model can handle variable mixed data and further improve the performance. Algorithm 1 outlines the training procedure of our proposed model.

Algorithm 1 Training procedure of our model

Initialize the model parameters $\boldsymbol{\theta}$ and a all-zero condition $\mathbf{G}^0$ for the first step
Given hyper parameters
- learning rate α, InterCTC loss weight λ

Loading pre-trained model or not
while Epoch < TotalEpoch **do**
  Given the input features $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ of a mixture speech and the corresponding transcriptions $\mathbf{Y} = \{\mathbf{Y}^1, \ldots, \mathbf{Y}^J\}$ of J different speakers
  Forward the $\text{Encoder}_{\text{mix}}$ with $\mathbf{X}$ and obtain the mixture hidden representations of $\mathbf{H}$ using Eq. (1)
  for $(i = 1; i < J; i{+}{+})$ **do**
    Concatenate the $\mathbf{H}$ with previously-estimated condition $\mathbf{G}^{i-1}$ and forward the LSTM layer as in Eq. (10)
    Forward the $\text{Encoder}_{\text{rec}}$ with the output of LSTM layer
    The output of $\text{Encoder}_{\text{rec}}$ is used to compute the Loss_{CTC} as well as determine the best permutation of transcriptions as in Eq. (4)
    The output of the intermediate layer in $\text{Encoder}_{\text{rec}}$ is used to compute the $\text{Loss}_{\text{InterCTC}}$ with above best transcription permutations
    $\mathbf{G}^i$ will also be regarded as the condition for the prediction of the next speaker
  end for
  Update the model using Eq. (9)
  Epoch = Epoch + 1
end while
return $\boldsymbol{\theta}$

4. Experiments

4.1. Setup

The proposed models are evaluated on two commonly used simulated multi-speaker speech datasets.

WSJ0-Mix. The dataset can be divided into two categories, namely the 2-speaker scenario and 3-speaker scenario. In the 2-speaker scenario, we use the common benchmark called WSJ0-2mix dataset introduced by [38] with a sampling rate of 16 KHz. The training and validation sets are generated by randomly selecting two utterances from different speakers from the WSJ0 si_tr_s partition, containing around 30 h and 10 h speech mixture, respectively. To mix the utterances, various signal-to-noise ratios (SNRs) are uniformly chosen from [0, 10] dB. For the test set, the mixture is similarly generated using utterances from the WSJ0 validation set si_dt_05 and evaluation set si_et_05, resulting in 5 h speech mixtures. For the 3-speaker experiments, similar methods are adopted except the number of speakers is three.

LibriMix. Our methods are additionally tested on LibriMix, a recent open-source dataset for multi-speaker speech processing. The LibriMix data is created by mixing the source utterances randomly chosen from different speakers in LibriSpeech [39] and the noise samples from WHAM! [40]. The SNRs of the mixtures are normally distributed with a mean of 0 dB and a standard deviation of 4.1 dB. LibriMix is composed of 2-speaker or 3-speaker mixtures, with or without noise conditions. For fast evaluation, we conducted our experiments on the train-100 subset from Libri2Mix, which contains around 100 h of 2-speaker mixture speech.

All the proposed models are implemented with ESPnet [41]. We followed the ESPnet recipe to set the hyperparameters of the model. For all Transformer- and Conformer-based models, EncoderMix is comprised of two CNN blocks and EncoderRec contains 8 Transformer or Conformer layers, depending on the model choices. For non-conditional chain models, the EncoderSD is a 4-layer Transformer or Conformer network, while the CondChain is a 1-layer LSTM network with 1024 hidden units. The common parameters of the Transformer and Conformer layers are: $d^{\text{head}} = 4, d^{\text{att}} = 256, d^{\text{ff}} = 2048$ for the number of heads, dimension of attention module, and dimension of feed-forward layer, respectively.

4.2. Results on WSJ0-Mix

In this part, we present the performance on the WSJ0-Mix corpus, which is shown in Table 1. To evaluate the effectiveness, we compare our conditional speaker chain based Conformer CTC model with a variety of systems including the hybrid systems, PIT-based end-to-end AR and NAR models, and conditional speaker chain based Transformer models. Since all PIT-based models are unable to deal with variable numbers of speakers, only the results of 2-speaker scenario are presented. To make a fair comparison with NAR methods, the end-to-end AR models are decoded only with greedy search.

For the PIT-based AR models, PIT-Conformer (5) shows the best performance, achieving a word error rate (WER) of 22.4% on the WSJ0-2mix test set. When comparing the NAR models, PIT-Transformer-CTC (6), which is only trained with

Table 1: *Word error rates (WERs) and real time factor (RTF) for multi-speaker speech recognition on WSJ0-Mix dataset. The RTF results are obtained by averaging the results of 5 decoding processes on CPUs.*

Models	**Training Data**	**WER (%)**		**RTF**
		WSJ0-2mix	**WSJ0-3mix**	
Hybrid model (w/ beam search)				
(1) PIT-DNN-HMM [24]	WSJ0-2mix	28.2	-	-
(2) DPCL + DNN-HMM [25]	WSJ0-2mix	16.5	-	-
E2E Autoregressive Model (w/ greedy search)				
(3) PIT-RNN [27]†	WSJ0-2mix	51.4	-	1.4293
(4) PIT-Transformer [33]†	WSJ0-2mix	37.0	-	1.4695
(5) PIT-Conformer	WSJ0-2mix	22.4	-	1.3970
E2E Non-autoregressive Model (w/ greedy search)				
(6) PIT-Transformer-CTC	WSJ0-2mix	50.3	-	0.1091
(7) Conditional-Transformer-CTC [33]†	WSJ0-2mix	41.0	-	0.1293
(8) Conditional-Transformer-CTC [33]†	WSJ0-1&2&3mix	29.4	53.3	-
(9) Conditional-Conformer-CTC	WSJ0-2mix	25.3	-	0.1824
+ hidden feature conditions	WSJ0-2mix	24.4	-	0.1758
+ InterCTC loss	WSJ0-2mix	22.3	-	0.1854
(10) Conditional-Conformer-CTC	WSJ0-1&2&3mix	23.4	39.1	0.1771 / 0.2096
+ hidden feature conditions	WSJ0-1&2&3mix	22.2	38.6	0.1741 / 0.2241
+ InterCTC loss	WSJ0-1&2&3mix	**19.9**	**34.3**	0.1732 / 0.2088

†: The results are obtained by the same implementation in [33] but w/o beam search and LM rescoring. When using both beam search and LM rescoring, the results are 14.9% / 37.9% of model (8) and 12.4% / 26.6% of model (10).

Table 2: *Word error rates (WERs) for multi-speaker speech recognition on LibriMix dataset.*

Models	**Dev**	**Test**
E2E Autoregressive Model (w/ greedy search)		
(1) PIT-Transformer	34.8	36.0
E2E Non-autoregressive Model (w/ greedy search)		
(2) PIT-Transformer-CTC	45.2	45.9
(3) Conditional-Transformer-CTC	32.7	33.3
(4) Conditional-Conformer-CTC + both	**24.5**	**24.9**

CTC loss, suffers a dramatic performance degradation (50.3%). There is no doubt that a pure CTC based encoder network can hardly model different speaker's speech simultaneously. When applying the conditional speaker chain based method, both model (7) and model (8) are better than PIT model. By combining the single and multi-speaker mixture speech, model (8) shows a significant improvement, whose WER is 29.5% on the WSJ0-2mix test set. For our conditional Conformer-CTC model (9), we explore two types of conditional features, including the "hard" CTC alignments and "soft" latent features after $\text{Encoder}_{\text{Rec}}$. Both approaches are better than above models with only a $\sim$0.07 seconds increase of latency and applying the "soft" features achieves a WER of 24.4%. By incorporating the intermediate loss, we can obtain a superior WER of 22.3%, reaching a strong AR PIT-Conformer model (5). However, after combining latent feature conditions and the intermediate CTC loss, we don't get a further improvement. Finally, we also train our model on the data of variable numbers of speakers and obtain the best WERs of 19.9% and 34.3%, which are even better than model (5) with only 1/7 latency.

We further investigate the correlation between the hypothesis generation order and the source signal length (from long to short), as shown in Table 3. We find that only 251/3000 utterances do not follow the order on 2-speaker scenario and the average Spearman's Coefficient is 0.833.

Table 3: *Correlation between the hypothesis (Hyp.) generation order and the source signal (Src.) length order on WSJ0-2mix.*

Hyp. \ **Src.**	long	short
1^{st} output	2749	251
2^{nd} output	251	2749

4.3. Results on LibriMix

The results on LibriMix are summarized in Table 2. From the table, we can see a quite similar trend as the WSJ0-Mix results in the previous subsection. Our Conditional-Conformer-CTC with both latent features conditions and intermediate CTC loss obtains the best WERs of 24.5% and 24.9% on dev and test sets, respectively, which yields up to 25% relative improvement compared with the Conditional-Transformer-CTC model.

5. Conclusions

In this study, we revisit our proposed conditional speaker chain based multi-speaker ASR by enhancing the NAR ability. Our improved model mainly includes a conditional speaker chain (CondChain) module and Conformer CTC based encoders. To boost the performance of a pure Conformer CTC encoder, we also investigate two approaches, which are using the "soft" latent features from the encoder output as speaker conditions and including an additional intermediate CTC loss. We evaluate the effectiveness of our model on two multi-speaker benchmarks, WSJ0-Mix and LibriMix. Our model shows consistent improvement over other models with only a slight increment of RTF and even better than a strong AR model in some cases.

6. References

[1] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, 2015.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.*, "Attention is all you need," in *Proc. NeurIPS*, 2017, pp. 5998–6008.

[3] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," in *Proc. ICASSP*. IEEE, 2016, pp. 4960–4964.

[4] L. Dong, S. Xu, and B. Xu, "Speech-Transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *Proc. ICASSP*. IEEE, 2018, pp. 5884–5888.

[5] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma *et al.*, "A comparative study on Transformer vs RNN in speech applications," in *Proc. ASRU*. IEEE, 2019, pp. 449–456.

[6] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*. ISCA, 2020, pp. 5036–5040.

[7] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi *et al.*, "Recent developments on ESPnet toolkit boosted by Conformer," in *Proc. ICASSP*. IEEE, 2021, pp. 5874–5878.

[8] J. Gu, J. Bradbury, C. Xiong, V. O. Li, and R. Socher, "Non-autoregressive neural machine translation," in *Proc. ICLR*, 2018.

[9] J. Libovický and J. Helcl, "End-to-end non-autoregressive neural machine translation with connectionist temporal classification," in *Proc. EMNLP*. ACL, 2018, pp. 3016–3021.

[10] J. Lee, E. Mansimov, and K. Cho, "Deterministic non-autoregressive neural sequence modeling by iterative refinement," in *Proc. EMNLP*. ACL, 2018, pp. 1173–1182.

[11] M. Stern, W. Chan, J. Kiros, and J. Uszkoreit, "Insertion Transformer: Flexible sequence generation via insertion operations," in *Proc. ICML*. PMLR, 2019, pp. 5976–5985.

[12] J. Gu, C. Wang, and J. Zhao, "Levenshtein Transformer," in *Proc. NeurIPS*, 2019, pp. 11 181–11 191.

[13] M. Ghazvininejad, O. Levy, Y. Liu, and L. Zettlemoyer, "Mask-predict: Parallel decoding of conditional masked language models," in *Proc. EMNLP-IJCNLP*. ACL, 2019, pp. 6112–6121.

[14] M. Ghazvininejad, O. Levy, and L. Zettlemoyer, "Semi-autoregressive training improves mask-predict decoding," *arXiv preprint arXiv:2001.08785*, 2020.

[15] C. Saharia, W. Chan, S. Saxena, and M. Norouzi, "Non-autoregressive machine translation with latent alignments," in *Proc. EMNLP*. ACL, 2020, pp. 1098–1108.

[16] N. Chen, S. Watanabe, J. Villalba, and N. Dehak, "Listen and fill in the missing letters: Non-autoregressive Transformer for speech recognition," *arXiv preprint arXiv:1911.04908*, 2019.

[17] 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*. ISCA, 2020, pp. 3655–3659.

[18] W. Chan, C. Saharia, G. Hinton, M. Norouzi, and N. Jaitly, "Imputer: Sequence modelling via imputation and dynamic programming," in *Proc. ICML*. PMLR, 2020, pp. 1403–1413.

[19] 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*. ISCA, 2020, pp. 5026–5020.

[20] Y. Higuchi, H. Inaguma, S. Watanabe, T. Ogawa, and T. Kobayashi, "Improved Mask-CTC for non-autoregressive end-to-end ASR," in *Proc. ICASSP*. IEEE, 2021, pp. 8363–8367.

[21] E. A. Chi, J. Salazar, and K. Kirchhoff, "Align-refine: Non-autoregressive speech recognition via iterative realignment," in *Proc. NAACL*. ACL, 2021, pp. 1920–1927.

[22] R. Fan, W. Chu, P. Chang, and J. Xiao, "CASS-NAT: CTC alignment-based single step non-autoregressive Transformer for speech recognition," *arXiv preprint arXiv:2010.14725*, 2020.

[23] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*. PMLR, 2006, pp. 369–376.

[24] Y. Qian, X. Chang, and D. Yu, "Single-channel multi-talker speech recognition with permutation invariant training," *Speech Communication*, vol. 104, pp. 1–11, 2018.

[25] T. Menne, I. Sklyar, R. Schlüter, and H. Ney, "Analysis of deep clustering as preprocessing for automatic speech recognition of sparsely overlapping speech." ISCA, 2019, pp. 2638–2642.

[26] H. Seki, T. Hori, S. Watanabe, J. L. Roux, and J. R. Hershey, "A purely end-to-end system for multi-speaker speech recognition," in *Proc. ACL*, 2018, pp. 2620–2630.

[27] X. Chang, Y. Qian, K. Yu, and S. Watanabe, "End-to-end monaural multi-speaker asr system without pretraining," in *Proc. ICASSP*. IEEE, 2019, pp. 6256–6260.

[28] X. Chang, W. Zhang, Y. Qian, J. Le Roux, and S. Watanabe, "End-to-end multi-speaker speech recognition with Transformer," in *Proc. ICASSP*. IEEE, 2020, pp. 6134–6138.

[29] N. Kanda, Y. Gaur, X. Wang, Z. Meng, Z. Chen, T. Zhou, and T. Yoshioka, "Joint speaker counting, speech recognition, and speaker identification for overlapped speech of any number of speakers," in *Proc. Interspeech*. ISCA, 2020, pp. 36–40.

[30] T. von Neumann, K. Kinoshita, L. Drude, C. Boeddeker, M. Delcroix, T. Nakatani, and R. Haeb-Umbach, "End-to-end training of time domain audio separation and recognition," in *Proc. ICASSP*. IEEE, 2020, pp. 7004–7008.

[31] T. von Neumann, K. Kinoshita, M. Delcroix, S. Araki, T. Nakatani, and R. Haeb-Umbach, "All-neural online source separation, counting, and diarization for meeting analysis," in *Proc. ICASSP*. IEEE, 2019, pp. 91–95.

[32] J. Shi, J. Xu, Y. Fujita, S. Watanabe, and B. Xu, "Speaker-conditional chain model for speech separation and extraction," in *Proc. Interspeech*. ISCA, 2020, pp. 2707–2711.

[33] J. Shi, X. Chang, P. Guo, S. Watanabe, Y. Fujita, J. Xu, B. Xu, and L. Xie, "Sequence to multi-sequence learning via conditional chain mapping for mixture signals," in *Proc. NeurIPS*, 2020, pp. 3735–3747.

[34] Y. Fujita, S. Watanabe, S. Horiguchi, Y. Xue, J. Shi, and K. Nagamatsu, "Neural speaker diarization with speaker-wise chain rule," *arXiv preprint arXiv:2006.01796*, 2020.

[35] J. Lee and S. Watanabe, "Intermediate loss regularization for ctc-based speech recognition," in *Proc. ICASSP*. IEEE, 2021, pp. 6224–6228.

[36] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Proc. ICASSP*. IEEE, 2017, pp. 4835–4839.

[37] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. ICASSP*. IEEE, 2017, pp. 241–245.

[38] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. ICASSP*. IEEE, 2016, pp. 31–35.

[39] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. ICASSP*. IEEE, 2015, pp. 5206–5210.

[40] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. L. Roux, "Wham!: Extending speech separation to noisy environments," in *Proc. Interspeech*. ISCA, 2019, pp. 1368–1372.

[41] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech*. ISCA, 2018, pp. 2207–2211.