Multi-Channel Transformer Transducer for Speech Recognition

Feng-Ju Chang, Martin Radfar, Athanasios Mouchtaris, Maurizio Omologo

Alexa Machine Learning, Amazon, USA

{fengjc, radfarmr, mouchta, omologo}@amazon.com

Abstract

Multi-channel inputs offer several advantages over single-channel, to improve the robustness of on-device speech recognition systems. Recent work on multi-channel transformer, has proposed a way to incorporate such inputs into end-to-end ASR for improved accuracy. However, this approach is characterized by a high computational complexity, which prevents it from being deployed in on-device systems. In this paper, we present a novel speech recognition model, *Multi-Channel Transformer Transducer (MCTT)*, which features end-to-end multi-channel training, low computation cost, and low latency so that it is suitable for streaming decoding in on-device speech recognition. In a far-field in-house dataset, our MCTT outperforms stagewise multi-channel models with transformer-transducer up to 6.01% relative WER improvement (WERR). In addition, MCTT outperforms the multi-channel transformer up to 11.62% WERR, and is 15.8 times faster in terms of inference speed. We further show that we can improve the computational cost of MCTT by constraining the future and previous context in attention computations.

Index Terms: Transducer, Transformer network, Attention layer, Multi-channel ASR, End-to-end ASR, Speech recognition, streamable ASR

1. Introduction

Voice assisted devices nowadays are usually equipped with multiple microphones for far-field speech recognition in noisy environments [1, 2]. By combining the spectral and spatial information of target and interference signals captured from different microphones, the beamforming approaches [3–8] have been demonstrated to benefit automatic speech recognition (ASR) systems substantially for improved recognition accuracy [6, 8, 9]. The beamformer thus has become the standard module, typically introduced before the ASR front-end and acoustic model.

The delay-and-sum and super-directive beamformers [10,11] are among the most popular beamforming methods for ASR, the latter one characterized by both its higher directivity and its lack of robustness to imperfect microphone arrays [12]. With the great success of deep neural networks, neural beamformers have gained significant interest and are becoming the state-of-the-art technologies in end-to-end all-neural ASR systems [13–22]. The neural beamforming methods are generally categorized into fixed beamforming (FBF) [18, 22] and adaptive beamforming (ABF) methods [13–17, 19, 21] depending on whether the beamforming weights are fixed or varied based on the input signals during inference time.

While neural beamforming approaches are attractive for their model capacity and direct access to the downstream ASR loss for optimizing the beamforming weights, their performance is still hindered by stagewise training. For example, the neural mask estimators in ABF methods [13, 14] usually must be pre-trained on synthetic data where the target speech and noise labels are well defined. The mismatch of these statistics between synthetic data and real-world data, however, can lead to noise leaking into the target speech statistics [23], and deteriorate its finetuning with the cascaded acoustic models.

Bypassing the need for stage-wise optimization and leveraging the core ability of transformer networks [24], i.e. attention on multiple modalities, a single integrated multi-channel transformer network was proposed [25] with both channel-wise and cross-channel attention layers for joint beamforming and acoustic modeling. Despite its effectiveness, this model is hard to apply to the streaming case such as on-device speech recognition [26], which demands low latency and low computation. First, it relies on an attention mechanism (encoder-decoder attention) over full encoder outputs to learn alignments between input and output sequences [27]. Second, the input audio is encoded in a bidirectional way, thus requiring a full utterance as input. Furthermore, the attention computation increases quadratically with the length of input sequences. Finally, the model size of the multi-channel transformer increases w.r.t. the number of microphones and the number of time frames [25] due to the use of affine transformations to aggregate multi-channel embeddings in cross-channel attention layers. For these reasons, it is unsuitable for on-device ASR systems with small memory.

There exist many streamable ways for alignment learning such as connectionist temporal classification (CTC) [28], transducer [29], monotonic chunkwise attention (MoChA) [30], and triggered attention [31], all of which can be integrated with transformer [24, 32–35]. In this work, we focus on transducer due to its outstanding performance over traditional hybrid models for streaming speech recognition [26, 36]. Several research efforts have combined transformer with transducer for single-channel speech recognition [37–40], but to the best of our knowledge, it is the first time that transducer is integrated with multi-channel transformer.

In addition to achieving streamable alignment learning, we further make the encoders streamable via limiting future context (right-context) and previous context (left-context) in both channel-wise and cross-channel attention computations for multi-channel audio encoding, and constraining previous context in self attention for output sequence embedding as well. For cross-channel attention computations, we also propose to use two simple combiners, the average and concatenation of multiple channels to create keys and values. In this way, our model size does not increase as the number of microphone and input sequence length increase.

In a far-field in-house dataset, we show that the proposed multi-channel transformer transducer outperforms single channel and stagewise neural beamformers cascaded with transformer transducers by 7.14% and 6.01% WERR respectively. Moreover, our model performs better than multi-channel transformer [25] up to 11.62% WERR and is 15.8 times faster in terms of inference speed (TP50). Finally, we improve the computational cost of both multi-channel audio encoder and label encoder for

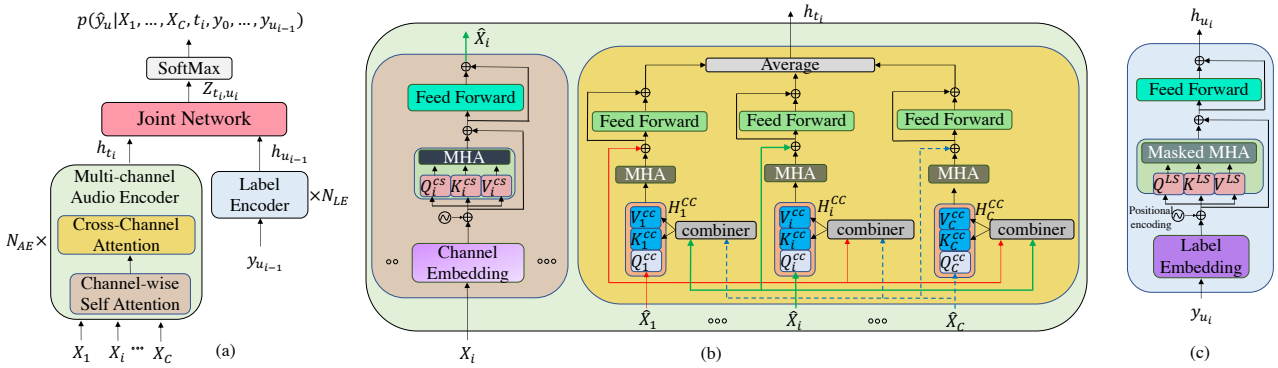

Figure 1: *An overview of the multi-channel transformer transducer (MCTT). (a) The high-level block diagram of MCTT (b) The multi-channel audio encoder architecture, which contains N_{AE} channel-wise attention layers (left) and cross-channel attention layers (right). MHA represents multi-head attention, and C is the number of channels. (c) The label encoder architecture, which consists of N_{LE} self-attention layers with token labels as inputs. Note that the layer norm is applied in both MHA and feed-forward layers, but omitted here.*

streaming case, by limiting both the left and right context in attention computations. Moreover, the performance gap between the causal attention and full attention versions of our model can be bridged by attending to a limited number of future frames.

2. Multi-Channel Transformer Transducer

2.1. Transducer

We denote C-channel of audio sequences as $\mathcal{X} = (X_1, ..., X_i, ..., X_C)$ where each channel is of T frames, $X_i = (\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, ..., \mathbf{x}_{i,T})$. We also denote a transcription label sequence of length U as $\mathbf{y} = (y_1, y_2, ..., y_U)$, where $y_u \in \mathcal{Z}$, and $\mathcal{Z}$ is a predefined set of token labels. As depicted in Fig. 1 (a), the transducer model encodes acoustic sequences first with a multi-channel audio encoder network (Fig. 1 (b)) to produce encoder output states as $h = (h_1, ..., h_T)$. For each encoder state h_t, the model predicts either a label or a blank symbol $\langle b \rangle$ with a joint network. If the model predicts a blank symbol, which indicates the lack of token label for that time step, then the model proceeds to the next encoder state. Different from CTC [28], the transducer model exploits not only the encoder output at time t but also the previous non-blank label history as inputs to predict the next output. The previously predicted labels are encoded with a label encoder as shown in Fig. 1 (c).

The transducer model defines a conditional distribution,

$$P(\hat{\mathbf{y}}|\mathcal{X}) = \prod_{i=1}^{T+U} P(\hat{y}_i|\mathcal{X}, t_i, y_0, ..., y_{u_{i-1}}) \quad (1)$$

where $\hat{\mathbf{y}} = (\hat{y}_1, ..., \hat{y}_{T+U}) \subset \{\mathcal{Z} \cup \langle b \rangle\}^{T+U}$ correspond to any possible alignment path with T blank symbols and U labels such that after removing all blank symbols in $\hat{\mathbf{y}}$ yields $\mathbf{y}$, and y_0 is the start of sentence symbol.

We can marginalize $P(\hat{\mathbf{y}}|\mathcal{X})$ over all possible alignments $\mathcal{A}(\mathcal{X}, \mathbf{y})$ to obtain the probability of the target label sequence $\mathbf{y}$ given the input multi-channel sequences $\mathcal{X}$,

$$P(\mathbf{y}|\mathcal{X}) = \sum_{\hat{\mathbf{y}} \in \mathcal{A}(\mathcal{X}, \mathbf{y})} P(\hat{\mathbf{y}}|\mathcal{X}) \quad (2)$$

This alignment probability summation can be computed efficiently with forward-backward algorithm [29].

2.2. Multi-Channel Audio Encoder

Previous work on the transducer framework [29, 37–40] relied only on single-channel input. To address multi-channel inputs, we propose to build our audio encoder based on multi-channel transformer network [25], as shown in Fig. 1 (b), containing two main blocks, channel-wise self-attention layers and cross-channel attention layers.

Channel-wise Self-Attention Layer (CSA): We start by projecting the source channel features (log-STFT magnitude and phase features are used in this work) to the dense embedding space for more discriminative representations. Then the embedded features plus the positional encoding [24] are fed into a set of learnable weight parameters to create Query (Q_i^{CS}), Key (K_i^{CS}), Value (V_i^{CS}). Similar to [25], the transformed features, Q_i^{CS} and K_i^{CS}, are used to compute the correlation across time steps within a channel via multi-head attention (MHA) [24]. The resulting attention matrix is then used to reweight the features of V_i^{CS} in each time step followed by a feed-forward network to produce the self-attention outputs.

Cross-Channel Attention Layer (CCA): Given the self-attended outputs per channel, the cross-channel attention layers aim to learn the contextual relationship across channels both within and across time steps. Inspired by [25], when we use the i-th channel to create Q_i^{CC}, the other channels are leveraged by a combiner to create K_i^{CC} and V_i^{CC}. Different from [25] which takes the sum of channel encodings after applying affine transformations (*Affine*), we investigate two simple combiners: (1) *Avg*: take the average of the other channels along both time and embedding axes, $H_i^{CC} = 1/C \sum_{j \neq i} \hat{X}_j$, which can be seen as the symmetric weight case of the *Affine* combiner in [25] (2) *Concat*: concatenate the other channels along the time axis, $H_i^{CC} = [\hat{X}_1; ...; \hat{X}_j; ...; \hat{X}_C]_{j \neq i}$. Here, $\hat{X}_j \in \mathbb{R}^{T \times d}$ and d is the embedding size. With this adaptation, the model parameters do not increase w.r.t. the number of microphones (C) and time frames (T) as in [25]. Finally, the cross-channel attention outputs are fused by a simple average.

2.3. Label Encoder and Joint Network

We leverage the transformer network to build the label encoder, as illustrated in Fig. 1 (c). An embedding layer converts previously predicted non-blank labels into vector representations. Then several linear layers project the embedding vectors in order to create Q^{LS}, K^{LS}, and V^{LS} followed by masked MHA

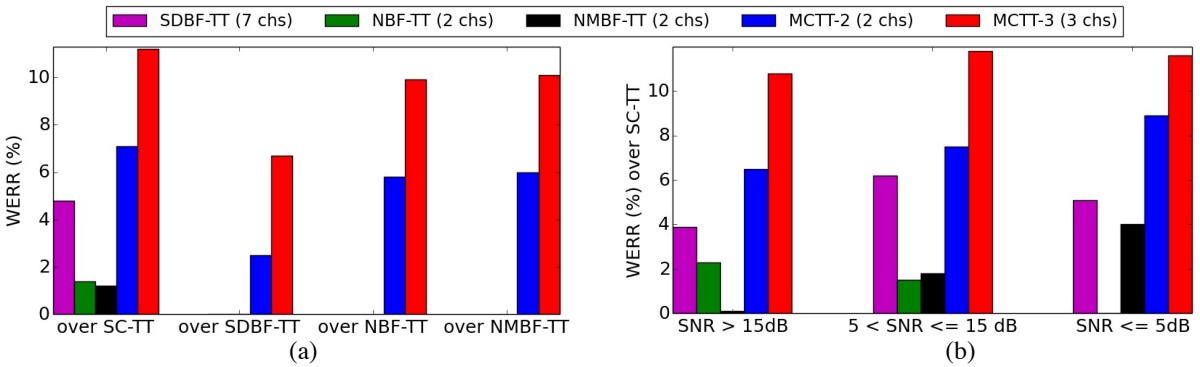

Figure 2: *The relative word error rate reduction, WERRs (%), by comparing the multi-channel transformer transducer (MCTT) to the beamfomers cascaded with transformer transducers. (a) WERRs over different methods (b) WERRs over SC-TT w.r.t. different SNR levels. A higher number indicates a better WER. Negative WERRs are not reported.*

Table 1: *The WERRs (%) of MCTT over MCT [25] for 2-channel inputs, and 3-channel inputs with different combiners.*

Method	Model Size (Million)	combiner	test-clean	test-other
MCT-2 [25]	18.59	-	0	0
MCTT-2	17.53	-	**11.62**	**4.51**
MCT-3 [25]	20.43	*Affine*	0	0
MCT-3 [25]	18.59	*Avg*	7.01	6.99
MCTT-3	17.53	*Avg*	**11.55**	**8.32**
MCTT-3	17.53	*Concat*	10.44	5.41

computations. The attention scores from the future frames are always masked out to ensure causality. Note that label encoder outputs do not attend to multi-channel audio encoder outputs, in contrast to the architecture in [25]. As discussed in Sec. 1, doing so poses a challenge for streaming applications. Instead, we use a joint network, which is a fully-connected feed-forward neural network with a single hidden layer and $tanh$ as the activation function. We concatenate outputs of multi-channel audio encoder and label encoder as inputs to the joint network.

2.4. Limiting History and Future Contexts in Attention

Attending to the whole input acoustic sequences in attention computations (i.e. full attention) not only disables the streaming inference but also gives the high computational complexity, $O(T^2)$ for computing encoder outputs. To reduce the computational cost and latency, we limit the left history frames (L) and future frames (R), $(\mathbf{x}_{t-L}, ..., \mathbf{x}_{t-R})$, of multi-channel encoder to compute $\mathbf{h}_{t_i}$. We also limit the left history frames (L) of the label encoder to compute $\mathbf{h}_{u_{i-1}}$. However, it also comes with potential performance drop, as investigated in experiments.

3. Experiments

3.1. Dataset

To evaluate our multi-channel transformer transducer (MCTT), we conduct a series of ASR experiments using over 2,200 hours of speech utterances from our in-house de-identified far-field dataset. The amount of training set, validation set (for model hyper-parameter selection), and test set are 2,000 hours, 24 hours, and 233 hours respectively. The device-directed speech data was captured using a smart speaker with 7 microphones, and a 63 mm aperture. The evaluation set has abundant annotations including

Table 2: *The inference speed comparisons of MCTT and MCT [25] in terms of Wall Clock Time (WCT).*

Method	Model Size (Million)	WCT (sec) TP50	TP90	TP99
MCT [25]	18.59	4.26	5.65	5.91
MCTT	17.53	**0.27**	**0.48**	**0.74**

the estimated SNR levels, and test-clean (no background speech) as well as test-other (with background speech) splits . In this dataset, 2 microphone signals of aperture distance and the super-directive beamformed signal by [10] using 7 microphone signals are employed through all the experiments.

3.2. Baselines

Following [25], one of the baselines is single channel + Transformer Transducer (SC-TT); we feed each of two raw channels individually into the transformer transducer for training and testing, and pick the best performed one. In addition, we compare to three stagewise beamforming methods cascaded with the transformer transducer (TT) models. The beamforming methods include Super-directive beamformer (SDBF) [10], Neural beamformer (NBF) [18], and Neural masked-based beamformer (NMBF) [13]. We denote the stagewise methods as SDBF-TT, NBF-TT, NMBF-TT, respectively. Note that SDBF-TT uses 7 microphone signals for beamforming as mentioned in section 3.1 while NBF-TT, NMBF-TT, and the proposed MCTT all take only 2 microphone signals as inputs. We also compare our method to multi-channel transformer network (MCT) [25], which is a single integrated multi-channel model.

3.3. Experimental Setup and Evaluation Metric

We set the number of audio encoder layers (N_{AE}=12) and label encoder layers (N_{LE}=6 for SC-TT, SDBF-TT, NBF-TT, NMBF-TT, N_{LE}=4 for MCT and MCTT) with 512 neurons to make all models with comparable number of parameters (18 millions), except for NMBF-TT (25.39 millions) due to the additional mask estimator [13]. Following [25], we use log-STFT square magnitude and phase features [41, 42] as inputs of our method, which are extracted every 10 ms with a window size of 25 ms from audio samples. The same setting is also applied to the feature extraction for baselines following [25]. The Adam optimizer [43], and subword tokenizer [44] with $4,001$ tokens are exploited. Results of all the experiments are reported as relative word error rate reduction (WERR) [25]. The higher the WERR is the better.

Table 3: *The WERRs (%) over full-attention MCTT (all contexts="inf") by limiting left context per layer for label encoder.*

MC Audio Mask		Label Mask	WERR (%)	
L	R	L	test-clean	test-other
inf	inf	inf	0	0
inf	inf	20	**2.10**	-0.24
inf	inf	4	1.17	**0.95**
inf	10	inf	-3.27	**-3.76**
inf	10	20	**-2.68**	-5.25
inf	10	4	-3.10	-4.00

Table 4: *The WERRs (%) over full audio attention based MCTT by limiting right context (R) per layer for MC audio encoder.*

MC Audio Mask		Label Mask	WERR (%)	
L	R	L	test-clean	test-other
inf	inf	20	0	0
inf	0	20	-23.65	-16.56
inf	2	20	-12.17	-8.04
inf	6	20	-5.99	**-2.74**
inf	10	20	**-4.88**	**-5.00**

3.4. Comparisons to Stagewise Multi-channel Models

We first compare the performance of MCTT with 2 channels, *Avg* combiner (MCTT-2) to the stagewise beamforming plus transformer transducer models, all with full attention audio encoder. The results are illustrated in Fig. 2. As shown in Fig. 2 (a), MCTT-2 outperforms SC-TT by 7.1% and neural beamformer + acoustic models (NBF-TT and NMBF-TT) by 6% in average. MCTT-2 also performs better than SDBF-TT by 2.48% even though it only considers 2 raw channels (2 chs). We further investigate if the super-directive beamformed signal is complementary to the other 2 channels by taking it as the third channel and feed them all to MCTT (denoted as MCTT-3). As can be seen in Fig. 2 (a), it provides 4% more improvements (WERRs) in average over all baselines as comparing to MCTT-2. In Fig. 2 (b), we further compare different methods w.r.t. different SNR levels. Again, we observe MCTT-2,3 achieve consistent improvements over SC-TT comparing to other methods across different SNRs.

3.5. Comparisons to Multi-channel Transformer

Next, we compare the proposed MCTT to MCT [25] with 2 channels and 3 channels (2 raw channels plus the super-directive beamformed signal) as inputs with different combiners. They are denoted as MCT-2,3 and MCTT-2,3 respectively. Note the combiner introduced in Sec. 2.2 is not needed for the 2-channel case, so its effect is only reported for the 3-channel case. We observe in Table 1 that MCTT-2 outperforms MCT-2 especially in test-clean split. Both MCT-3 and MCTT-3 with *Avg* combiner perform better than MCT-3 with *Affine* combiner, and MCTT-3 performs the best. Besides, using *Avg* combiner is more effective than using *Concat* combiner.

We further evaluate inference speed by measuring decoding time over 10,000 utterances on a Intel Xeon® Platinum 8175M processors machine using 1 CPU per method to process an utterance at a time with greedy search decoding. The Top Percentile values, TP50 (median), TP90, and TP99 wall clock times (WCT) are shown in Table 2. Most of inference time of MCT has been dedicated to the encoder-decoder attention, while MCTT does not have this issue and achieves 15.4 times faster inference speed in terms of TP50.

Table 5: *The WERRs (%) over full audio attention based MCTT (with left context of label encoder L=20) by limiting both MC audio contexts (L and R) and label contexts (L) for streaming.*

MC Audio Mask		Label Mask	WERR (%)	
L	R	L	test-clean	test-other
inf	inf	20	0	0
20	0	20	-27.68	-22.87
20	10	20	**-7.37**	**-6.79**
20	20	20	**-7.54**	**-6.31**
10	0	20	-24.08	-22.45
10	10	20	-9.51	-11.08
10	20	20	**-7.28**	**-7.92**

3.6. Results of Limiting Contexts in Attention Computation

Finally, we ran training and decoding experiments using MCTT with limited attention windows over audio and text labels, with a view to build streaming multi-channel (MC) speech recognition systems with low latency and low computation cost. "inf" in Table 3, 4, 5 means we employ all of the left or right contexts. Besides, MC Audio Mask, and Label Mask indicate the coverage of audio/label frames to be considered in attention of Multi-Channel audio encoder and label encoder respectively.

We start from evaluating how the left context of the label encoder affects performance. In Table 3, we show that constraining each layer to use only 4 previous label frames yields the similar accuracy with the model using all previous frames per layer (1.06% WERR in average when MC audio mask R=inf). As constraining right context of MC audio to 10, the WERR differences are also small; the maximum WERR difference is 0.24% (-3.76%-(-4)%) when compared to using all previous frames per layer. It indicates that very limited left context for label encoder is good enough for MCTT.

We then fix the left context of label encoder to 20 , and constrain the MC audio encoder to attend to only the left of the current frame (so that no latency is introduced). As shown in Table 4, the WERs drastically degrade by 23.65% and 16.56% in test-clean and test-other splits comparing to MCTT with full attention MC audio encoder. By allowing the model to see some future frames (e.g. $R = 10$), we can bring down the WER degradation to $\leq 5\%$ for both splits.

Table 5 reports the results when limiting both the left and right contexts of MC audio encoder. By doing so, not only the latency can be reduced, but also the time complexity for one-step inference becomes a constant. We limit the left context of MC audio encoder to 20 and 10 respectively, and then increase right context from 0 to 20. As can be seen in both cases, with the look-ahead to few future frames (e.g. $R = 20$), the WER gap to the full-attention audio encoder based model was narrowed down to 6.31% and 7.92% respectively in test-other split.

4. Conclusion

We propose a novel speech recognition model, Multi-Channel Transformer Transducer, which is capable of leveraging multi-channel inputs in an end-to-end fashion and applicable to streaming decoding for speech recognition. We show that the proposed MCTT outperforms its stagewise counterparts, and significantly reduces the inference time against multi-channel transformer [25]. Furthermore, by limiting the left contexts and with look-ahead to few future frames, we can not only improve the computation cost, but also bridge the gap between the performance of left-only attention and full attention models.

5. References

[1] R. Haeb-Umbach, J. Heymann, L. Drude, S. Watanabe, M. Delcroix, and T. Nakatani, "Far-field automatic speech recognition," *Proceedings of the IEEE*, 2020.

[2] E. Vincent, T. Virtanen, and S. Gannot, *Audio source separation and speech enhancement*. John Wiley & Sons, 2018.

[3] M. Omologo, M. Matassoni, and P. Svaizer, "Speech recognition with microphone arrays," in *Microphone arrays*. Springer, 2001, pp. 331–353.

[4] M. Wölfel and J. McDonough, *Distant speech recognition*. John Wiley & Sons, 2009.

[5] K. Kumatani, J. McDonough, and B. Raj, "Microphone array processing for distant speech recognition: From close-talking microphones to far-field sensors," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 127–140, 2012.

[6] K. Kinoshita *et al.*, "A summary of the reverb challenge: state-of-the-art and remaining challenges in reverberant speech processing research," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, pp. 1–19, 2016.

[7] T. Virtanen, R. Singh, and B. Raj, *Techniques for noise robustness in automatic speech recognition*. John Wiley & Sons, 2012.

[8] T. Menne, J. Heymann, A. Alexandridis, K. Irie, A. Zeyer, M. Kitza, P. Golik, I. Kulikov, L. Drude, R. Schlüter, H. Ney, R. Haeb-Umbach, and A. Mouchtaris, "The rwth/upb/forth system combination for the 4th chime challenge evaluation," in *CHiME-4 workshop*, 2016.

[9] J. Barker, R. Marxer *et al.*, "The third 'chime'speech separation and recognition challenge: Dataset, task and baselines," in *ASRU*, 2015.

[10] S. Doclo and M. Moonen, "Superdirective beamforming robust against microphone mismatch," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 2, pp. 617–631, 2007.

[11] I. Himawan, I. McCowan, and S. Sridharan, "Clustered blind beamforming from ad-hoc microphone arrays," *TASLP*, vol. 19, no. 4, pp. 661–676, 2010.

[12] X. Chen, J. Benesty, G. Huang, and J. Chen, "On the robustness of the superdirective beamformer," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 838–849, 2021.

[13] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *ICASSP*, 2016.

[14] H. Erdogan, J. R. Hershey *et al.*, "Improved mvdr beamforming using single-channel mask prediction networks." in *Interspeech*, 2016.

[15] T. Ochiai, S. Watanabe *et al.*, "Multichannel end-to-end speech recognition," *arXiv preprint arXiv:1703.04783*, 2017.

[16] X. Chang, W. Zhang, Y. Qian, J. Le Roux, and S. Watanabe, "Mimo-speech: End-to-end multi-channel multi-speaker speech recognition," in *ASRU*, 2019.

[17] ——, "End-to-end multi-speaker speech recognition with transformer," in *ICASSP*, 2020.

[18] K. Kumatani, W. Minhua, S. Sundaram, N. Ström, and B. Hoffmeister, "Multi-geometry spatial acoustic modeling for distant speech recognition," in *ICASSP*, 2019.

[19] B. Li *et al.*, "Neural network adaptive beamforming for robust multichannel speech recognition," in *Interspeech*, 2016.

[20] X. Xiao, S. Watanabe *et al.*, "Deep beamforming networks for multi-channel speech recognition," in *ICASSP*, 2016.

[21] Z. Meng, S. Watanabe *et al.*, "Deep long short-term memory adaptive beamforming networks for multichannel robust speech recognition," in *ICASSP*, 2017.

[22] Y. Liu, P. Zhang, and T. Hain, "Using neural network front-ends on far field multiple microphones based speech recognition," in *ICASSP*, 2014.

[23] L. Drude, J. Heymann, and R. Haeb-Umbach, "Unsupervised training of neural mask-based beamforming," *arXiv preprint arXiv:1904.01578*, 2019.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurNIPS*, 2017.

[25] F.-J. Chang, M. Radfar, A. Mouchtaris, B. King, and S. Kunzmann, "End-to-end multi-channel transformer for speech recognition," *ICCASP*, 2021.

[26] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang *et al.*, "Streaming end-to-end speech recognition for mobile devices," in *ICASSP*, 2019.

[27] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[28] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[29] A. Graves, "Sequence transduction with recurrent neural networks," *ICML workshop*, 2012.

[30] C.-C. Chiu and C. Raffel, "Monotonic chunkwise attention," *ICLR*, 2018.

[31] N. Moritz, T. Hori, and J. Le Roux, "Triggered attention for end-to-end speech recognition," in *ICASSP*, 2019.

[32] L. Dong, S. Xu, and B. Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *ICASSP*, 2018.

[33] L. Lu, C. Liu, J. Li, and Y. Gong, "Exploring transformers for large-scale speech recognition," *arXiv preprint arXiv:2005.09684*, 2020.

[34] Y. Wang *et al.*, "Transformer-based acoustic modeling for hybrid speech recognition," in *ICASSP*, 2020.

[35] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[36] T. N. Sainath, Y. He, B. Li, A. Narayanan, R. Pang, A. Bruguier, S.-y. Chang, W. Li, R. Alvarez, Z. Chen *et al.*, "A streaming on-device end-to-end model surpassing server-side conventional model quality and latency," in *ICASSP*, 2020.

[37] Z. Tian, J. Yi, J. Tao, Y. Bai, and Z. Wen, "Self-attention transducers for end-to-end speech recognition," *Interspeech*, 2019.

[38] C.-F. Yeh, J. Mahadeokar, K. Kalgaonkar, Y. Wang, D. Le, M. Jain, K. Schubert, C. Fuegen, and M. L. Seltzer, "Transformer-transducer: End-to-end speech recognition with self-attention," *arXiv preprint arXiv:1910.12977*, 2019.

[39] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, "Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss," in *ICASSP*, 2020.

[40] W. Huang, W. Hu, Y. T. Yeung, and X. Chen, "Conv-transformer transducer: Low latency, low frame rate, streamable end-to-end speech recognition," *Interspeech*, 2020.

[41] Z.-Q. Wang and D. Wang, "Combining spectral and spatial features for deep learning based blind speaker separation," *TASLP*, vol. 27, no. 2, pp. 457–468, 2018.

[42] Z.-Q. Wang, J. Le Roux, and J. R. Hershey, "Multi-channel deep clustering: Discriminative spectral and spatial embeddings for speaker-independent speech separation," in *ICASSP*, 2018.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[44] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *ACL*, 2016.