INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

End-to-End Neural Speaker Diarization with Absolute Speaker Loss

Chao Wang, Jie Li, Xiang Fang, Jian Kang, Yongxiang Li

China Telecom Corporation Ltd. Data&AI Technology Company
{wangc75,lij86,fangx1,kangj30,liyx25}@chinatelecom.cn

Abstract

End-to-end neural speaker diarization (EEND) has proved to be a very promising method in speaker diarization, especially in tackling overlapping speech recordings. In this paper, we propose a new approach to EEND that incorporates an absolute speaker loss function, which can force the network to consider global speaker identity information in the training phase, and keeps one-stage inference at the same time. Besides, we modify the pre-processing module and do not need feature splice, which results in longer contextual information and supports longer recording input when inferencing. As a result, with our proposed one-stage system, we achieve better results in simulated librispeech conversation-like data sets compared to EEND-VC, a two-stage system. We evaluate our experiments in different chunkings, different durations and different overlap ratios, and achieve up to 70% relative improvement in terms of DER over baseline EEND-VC on short recordings and up to 7.5% on long recordings.

Index Terms: speaker diarization, end-to-end, neural networks.

1. Introduction

The goal of a speaker diarization system is to estimate the temporal boundary of each talking speakers in real audio recordings [1, 2]. An accurate diarization result is crucial for applications such as meeting summarization, turn-taking analysis of telephone conversations and so on. Recent diarization technologies can be categorized into two approaches: cluster-based systems and end-to-end neural approaches.

The well-established cluster-based diarization systems rely on embedding extractors and clustering algorithms. In general, those approaches first train a network to get speaker embeddings from audio streams. Commonly used speaker embeddings include i-vectors [3], d-vectors [4, 5], and x-vectors [6]. Then in the test phase, the recording is segmented into short homogeneous blocks and the speaker embeddings are computed for each block. In most cases, an assumption is made that "only one speaker is active in each block". Finally, the speaker embeddings are clustered into several centers, from which the time intervals and corresponding speaker labels are obtained. Various networks [7, 8] and techniques to extract and cluster the embeddings have been explored for speaker diarization tasks in [9, 10, 11]. There is a clear disadvantage of the cluster-based systems that they can hardly process the overlapping speech due to the hypothesis that only one speaker is active in a specific window. Besides, they rely on multiple modules and cannot be optimized to minimize diarization errors directly because the clustering is performed in an unsupervised manner, this also make them a two-stage system.

Recent approaches have sought to address these limitations by incorporating end-to-end neural networks. These systems directly output frame-level predictions without clustering. EEND [12, 13, 14] model the diarization problem by using the Permutation Invariant Training(PIT[15, 16]) criterion, and frame-level outputs and binary cross-entropy function make it possible to handle overlapped speech. However, the utterance-level PIT in EENDs ignoring the global identity information of speakers, and the ability to extend to a flexible number of speakers is limited. Besides, it is hard to apply EENDs to long audio recordings [17] (e.g. audio duration longer than 10 minutes) because of poor generalization to the long data and the CPU memory constraint. Various works have been done to explore the improvement of EEND. In EEND-VC [17, 18, 19], the authors tried to use a learnable global speaker embedding dictionary to achieve better performance, however, they make EEND a two-stage approach, one stage for predicting the speaker labels, the other stage for clustering.

In this paper, we propose a simple but effective approach to EEND that is much better than EEND-VC while keeping one-stage inference. The contribution of our work includes two key points. Firstly, the absolute speaker loss is designed to force the network to learn global and absolute speaker information (i.e. the frame-level global speaker embeddings) during the training process. Secondly, the lightweight pre-processing network makes it possible to handle longer recordings with lower memory consumption. We called our system EEND-ASL (Absolute Speaker Loss). Based on this system, we achieve up to 70% relative improvement in terms of DER over baseline EEND-VC on short recordings and up to 7.5% on long recordings.

The rest of the paper is organized as follows. First, we introduce the related works and our method in section 2 and 3. Then in section 4, the effectiveness of the proposed framework is evaluated. Finally, we conclude the paper in section 5.

2. Baseline System

In this section, we present an overview of the self-attentive end-to-end diarization model (EEND) [13] and EEND-vector clustering (EEND-VC) [17].

2.1. EEND

Let $X = (\mathbf{x}_t \in \mathbb{R}^F | t = 1, ..., T)$ be the input observation sequence feature at time step t. $Y = (\mathbf{y}_t | t = 1, ..., T)$ and $\mathbf{y}_t = [y_{t,c} \in \{0, 1\} | c = 1, ...C]$ be the relative ground-truth speaker label, $S = (\mathbf{s}_t | t = 1, ..., T)$ and $\mathbf{s}_t = [s_{t,n} \in \{0, 1\} | n = 1, ...N]$ be the absolute ground-truth speaker label sequence at time step t, of which C means the total number of speakers at current recording, and N means the total number of speakers at the training data set. In the EEND framework, we have the following formulas to compute the estimated output:

10.21437/Interspeech.2023-656

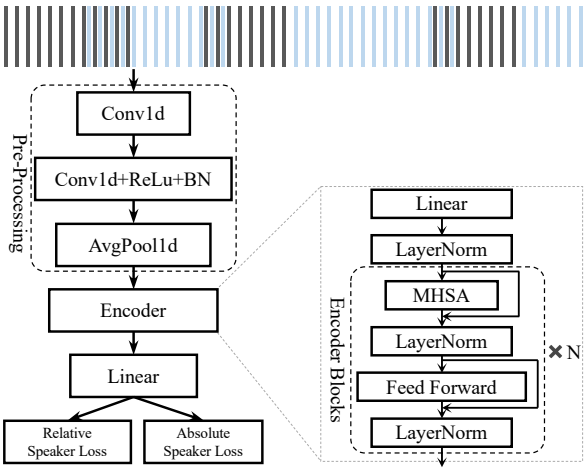

Figure 1: *The overall framework of our proposed diarization system. The entire network consists of two loss functions, relative speaker loss function and absolute speaker loss function. Furthermore, the modified Pre-Processing module is also a key component in our network.*

$$\begin{aligned} H &= (\mathbf{h}_t \in \mathbb{R}^D | t = 1, ..., T) \\ &= \mathbf{Encoder}(X) \in \mathbb{R}^{D \times T} \\ \hat{Y} &= (\hat{\mathbf{y}}_t \in \mathbb{R}^C | t = 1, ..., T) \\ &= \sigma(\mathbf{Linear}(H)) \in \mathbb{R}^{C \times T} \end{aligned} \tag{1}$$

where **Encoder** means the Encoder layer and σ means **Sigmoid** function. Then the PIT loss (in this paper we called it relative speaker loss) is written as follows:

$$L^R = \frac{1}{TC} \min_{\phi \in perm(C)} \sum_t \mathbf{BCE}(\mathbf{l}_t^\phi, \hat{\mathbf{y}}_t) \tag{2}$$

where $perm(C)$ is the set of all the possible permutations of $1, ..., C$ and $\mathbf{l}_t^\phi$ is the ϕ-th permutation of the relative label.

2.2. EEND-VC

In EEND-VC [17], the authors tried to use global speaker information to formula an extra speaker embedding loss. After estimating the diarization results, for the purpose of solving the inter-block permutation problem, they estimate the speaker embedding, $\hat{e}_s$, corresponding to the diarization result of the s-th speaker. Then, the speaker embedding loss is written as follows.

$$\begin{aligned} L^{speaker} &= -\log \left(\frac{\exp(-d(E_s, \hat{e}_s))}{\sum_{n=1}^N \exp(-d(E_n, \hat{e}_s))} \right) \\ d(E_n, \hat{e}_s) &= \alpha ||E_n - \hat{e}_s||^2 + \beta \end{aligned} \tag{3}$$

where E_n is a learnable variance-normalized global speaker embedding associated with the n-th training speaker.

3. Proposed

In this section, we will describe our overall framework. As shown in Figure 1, the entire network consists of two loss functions, relative speaker loss (also known as PIT loss) function and absolute speaker loss function. Furthermore, the modified Pre-Processing module is also a key component in our network.

3.1. Absolute Speaker Loss

Inspired by [20] and circle loss [21], we designed absolute speaker loss. Similar with equation 1, we got the prediction score $\hat{S}$ after the Linear layer:

$$\begin{aligned} H &= (\mathbf{h}_t \in \mathbb{R}^D | t = 1, ..., T) \\ &= \mathbf{Encoder}(\mathbf{PP}(X)) \in \mathbb{R}^{D \times T} \\ \hat{S} &= (\hat{\mathbf{s}}_t \in \mathbb{R}^N | t = 1, ..., T) \\ &= \mathbf{Linear}(H) \in \mathbb{R}^{N \times T} \end{aligned} \tag{4}$$

We treat the diarization problem as a special case of multi-label classification, that is, one frame corresponds to at least one speaker. We noticed that the class-level labels and pair-wise labels are determined when the absolute speaker label in the whole datasets is provided. let $p_i = \hat{\mathbf{s}}_t^i, i \in \mathbb{N}$ and $p_j = \hat{\mathbf{s}}_t^j, j \in \mathbb{P}$ be the i-th speaker and j-th speaker prediction score at frame t, where $\mathbb{N}$ and $\mathbb{P}$ are negative and positive sample set, respectively. Let p_0 be the threshold prediction score for an additional speaker $\mathbf{0}$. Similar to [20] and circle loss[21], we hope all positive scores to be greater than all negative scores, as well as all positive scores to be greater than p_0 and all negative scores to be less than p_0, so we formula our absolute loss as follows:

$$\begin{aligned} L^A &= log \left(1 + \sum_{i \in \mathbb{N}, j \in \mathbb{P}} e^{p_i - p_j} + \sum_{i \in \mathbb{N}} e^{p_i - p_0} + \sum_{j \in \mathbb{P}} e^{p_0 - p_j} \right) \\ &= log \left(e^{p_0} + \sum_{i \in \mathbb{N}} e^{p_i} \right) + log \left(e^{-p_0} + \sum_{j \in \mathbb{P}} e^{-p_j} \right) \end{aligned} \tag{5}$$

When we set the threshold prediction score of the additional speaker to 0, i.e., $p_0 = 0$, we got:

$$L^A = log \left(1 + \sum_{i \in \mathbb{N}} e^{p_i} \right) + log \left(1 + \sum_{j \in \mathbb{P}} e^{-p_j} \right) \tag{6}$$

Thus, we finally got our absolute speaker loss. Different from the learnable global speaker embedding dictionary in EEND-VC, we use the absolute speaker label (i.e., the global speaker identity in the whole data set) as our global speaker information.

3.2. Pre-Processing

As mentioned above, the lightweight pre-processing network is crucial to handle longer recordings with lower memory consumption. We use convolution layers instead of feature splicing. The first Conv1d in the pre-processing module has a convolution kernel size of 15 in time steps, which is the same as the second Conv1d, thus the network has a receptive field of total 29 in time steps (i.e., we got 14 time steps of contextual information in left and right respectively, while the feature splicing only got 7 time steps contextual information in left and right respectively). We adopt AvgPool1d with a stride of 10 as our down-sampling layer to increase the contextual information and receptive field of convolution. The details of the encoder was also shown in Figure 1. Similar to [13] and [22], the configuration of the encoder block consists of two sub-layers, the first is a multi-head self-attention layer, and the second is a feed-forward layer. In this work, we have $N = 6$ blocks. Our network was implemented through PyTorch.

Table 1: *DERs (%) of the EEND-VC and the proposed models for each test set that differs in the chunking and duration*

Model	Chunking size (seconds)				Clustering	test data duration (minutes)			
	30	200	1000	1200		3	5	10	20
EEND-VC	✓					8.80	8.90	9.06	8.94
EEND-VC		✓				7.16	7.64	9.77	10.01
EEND-VC			✓			6.58	7.45	8.42	9.42
EEND-VC				-		-	-	-	-
EEND-VC	✓				✓	4.49	4.18	4.37	4.40
Proposed w/o ASL	✓					9.48	9.61	10.25	9.81
Proposed w/o ASL		✓				3.85	5.61	7.29	7.66
Proposed w/o ASL			✓			**1.21**	**1.21**	1.63	5.88
Proposed w/o ASL				✓		**1.21**	**1.21**	1.43	5.82
Proposed	✓					**6.09**	**6.26**	**6.20**	**5.86**
Proposed		✓				**3.04**	**4.49**	**5.07**	**5.11**
Proposed			✓			1.24	1.26	**1.63**	**3.87**
Proposed				✓		1.24	1.26	**1.42**	**4.07**
Proposed				✓	✓	1.24	1.26	1.43	4.15

3.3. Training Objectives

Now, the two losses, within-recording relative speaker loss and between-recording absolute speaker loss, are jointly optimized using a mixing parameter λ:

$$L^{TOTAL} = (1-\lambda)L^{R} + \lambda L^{A} \tag{7}$$

where λ is a hyperparameter to balance the absolute speaker loss and relative speaker loss. According to our preliminary experiments, we observe λ slightly affects the final results when the training dataset is unbalanced (i.e., the number of utterances of the different speakers is unbalanced). A reasonable range of values for lambda is $[0.1, 0.5]$. In this paper, we set λ to 0.1.

4. Experiments

We evaluate the effectiveness of our method on simu-librispeech data sets compared to EEND-VC[17]. We conduct experiments in different chunkings, different test recording durations and different overlap ratios, the results show that the proposed ASL can deal with overlapping recordings well, and we have a better and more stable performance in DER.

4.1. Data

Librispeech data sets[23] are used in our experiments to simulate a conversation-like mixtures of two speakers. The mixture simulation algorithm we used in our work is from [12] [1]. Same as [12], we performed offline data augmentation by using MUSAN [24] and RIR_NOISES[25] during simulation. In total, we generated 50k mixtures, about 5545 hours of training data sets and 4 different test data sets, each test set contains 500 recordings and the average duration of recordings is 3, 5, 10, and 20 minutes, respectively. Finally, we randomly selected 10k training mixtures (roughly 1110 hours, 2680 speakers) from all 50k training mixtures. This data sets is called simu-librispeech.

4.2. Network Configuration

We applied a hamming window and got 23-dimensional log-Mel filterbank features with a frame length of 25ms and a frame shift of 10ms. Online data reverberation and noise are applied during the training process. Unlike EEND and EEND-VC, we do not require feature splice (i.e., a single frame is spliced using its left context and right context frames), which will reduce some memory consumption. In addition, we perform all operations include the compute of acoustic features and online data augmentation on GPU in our network.

For both EEND-VC and proposed systems, we used the same network architecture as [13]. For Encoder, we used 6 blocks with 256 attention units containing 8 heads. Noam scheduler [22] and Adam optimizer are adopted and the increasing steps (warm-up steps) were set to 40% of the total steps. The batch size was 72 on 4 Tesla V100 and the number of training epochs was 100. After training, we obtained the final model by averaging the weights over the last 10 epochs.

In the training phase, the training objectives L^{TOTAL} in equation 7 are optimized, while in the test phase, we do not use the absolute part and just predict the relative speaker labels.

4.3. Results

Table 1 shows the diarization error rates, DERs (%) of the EEND-VC, and the proposed models for each test sets that differs in the chunking and duration, where DER is the sum of three different error types: missed detection (MI), false alarm (FA), and speaker confusion (CF). The results reveals the robustness of our EEND-ASL in intra-block and inter-block. We also tried to cluster our results, but the performance did not get better, because the ASL has already plays a role of clustering during the training phase. Figure 2 also shows that.

4.3.1. Different Duration

[17] has proved that EEND-VC outperforms EEND, so we only focus on comparing the performance of EEND-VC with our proposed model. Let's consider two scenarios: short chunking size and long chunking size. In the case of a short chunking size (e.g., 30 seconds), almost all test recordings are split into blocks by chunking size, the EEND-ASL has a better and relatively stable performance, which reveals the robustness of our proposed method in inter-block variance. When the chunking size becomes longer (e.g. chunking size is 1000 seconds) and

[1] https://github.com/hitachi-speech/EEND

Table 2: *DERs (%) of different systems for each overlap ratio.*

Model	overlap (%)		
	0-30	30-60	60-90
EEND-VC	4.80	4.60	3.31
Proposed w/o ASL	10.38	3.81	**1.38**
Proposed	**4.61**	**3.04**	1.51

Table 3: *DERs (%) detail performance analysis for each test set. MI, FA, and CF indicate the missed detection, false alarm and speaker confusion errors respectively.*

Model	Errors	test data duration (minutes)			
		3	5	10	20
EEND-VC	MI	2.40	2.09	2.32	2.26
EEND-VC	FA	1.51	1.51	1.46	1.47
EEND-VC	CF	0.52	0.59	0.58	0.67
Proposed w/o ASL	MI	0.58	0.59	0.81	1.00
Proposed w/o ASL	FA	0.49	0.51	0.53	0.92
Proposed w/o ASL	CF	0.14	0.10	0.09	3.90
Proposed	MI	0.48	0.70	0.68	0.89
Proposed	FA	0.65	0.48	0.69	0.75
Proposed	CF	0.11	0.08	0.05	2.43

much larger than the test duration, the EEND-ASL has excellent performance, which reveals the robustness of our proposed method in intra-block variance.

As the test duration increases, the performance of our model decreases slightly, but still much better than EEND-VC. Besides, the results between the proposed with and without ASL (i.e., the ablation experiment) show that we have relatively stable and better performance in different chunking sizes. Overall, we achieved the best results on almost all test sets.

4.3.2. Different Chunking Size and Overlap Ratio

As we can see, for each test set, all systems in Table 1 have a performance degradation (especially the proposed without ASL) when using short chunking size because of the inter-block label permutation problem. Obviously, the proposed EEND-ASL has a relatively smaller performance degradation and maintains a more stable performance, which reveals the robustness of our proposed method in inter-block. Besides, we can find that the EEND-VC has a limitation of chunking size (i.e., up to 1000 seconds) in a one-stage process because of poor generalization to the long data and the CPU memory constraints. Thanks to the use of the pre-precessing module, we mitigated this limitation and reached up to 1200 seconds.

Table 2 shows the DERs in each overlap condition. We combine all 2000 recordings test data sets and categorized them into several overlap ratio ranges and obtained DER in each condition. We can conclude that our system has a better DER, especially in large overlapping ratio, which shows that the EEND-ASL handles overlapping speech well compared to EEND-VC.

4.3.3. Detailed Analysis

The numbers reported in Table 3 shows the detailed performance analysis for each test set that differs in the chunking and duration. Benefit from the absolute speaker loss, we achieve the lowest confusion error in short recordings. Figure 2 shows the t-SNE visualization of the speaker embeddings of the 15 test speakers. We averaged the frame-level speaker embeddings of utterances into segments-level speaker embeddings. It clearly shows distinguished clusters for each speaker compared to the proposed without ASL, which proves that we can estimate the global speaker embeddings accurately and that the ASL has already plays a role of clustering during training phase.

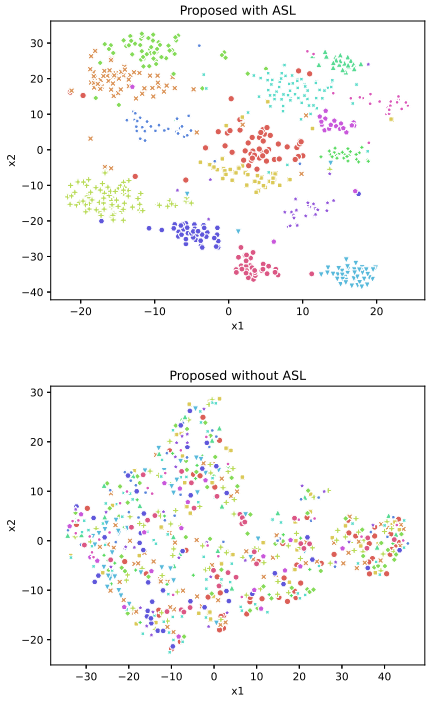

Figure 2: *t-SNE of the 15 speaker's embeddings. The speakers were randomly selected from Librispeech clean test set. Top and bottom refer to the proposed with and without ASL, respectively.*

In conclusion, by using the global absolute speaker identity, ASL force the network to formula a multi-label classification problem, thus the bottle-neck embedding of each frame could be able to represent the absolute speakers in a certain receptive field. The representation will help the optimizer to learn a more discriminative network, and with the continuous training, the representation becomes more and more accurate. Overall, the ASL encourage the optimizer to learn more accurate representations and a more discriminative network.

5. Conclusions

In this paper, we propose a simple but effective approach to EEND that incorporates an absolute speaker loss function, called EEND-ASL(Absolute Speaker Loss), and we get better results with our proposed one-stage system than with a two-stage system. Besides, the lightweight pre-processing network makes it possible to handle longer recordings with lower memory consumption. Furthermore, the proposed ASL can deal with overlapping speech recordings, and the experiments in simu-librispeech data sets reveal the robustness of our EEND-ASL in intra-block and inter-block. Finally, we achieve up to 70% relative improvement over baseline EEND-VC on short recordings and up to 7.5% on long recordings.

6. References

[1] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 356–370, 2012.

[2] T. J. Park, N. Kanda, D. Dimitriadis, K. J. Han, S. Watanabe, and S. Narayanan, "A review of speaker diarization: Recent advances with deep learning," *Computer Speech & Language*, vol. 72, p. 101317, 2022.

[3] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[4] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4052–4056.

[5] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5115–5119.

[6] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *2016 IEEE Spoken Language Technology Workshop (SLT)*, 2016, pp. 165–170.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[8] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Interspeech 2020*. ISCA, 2020.

[9] M. Díez, F. Landini, L. Burget, J. Rohdin, A. Silnova, K. Žmolíková, O. Novotný, K. Veselý, O. Glembek, O. Plchot, L. Moner, and P. Matejka, "BUT System for DIHARD Speech Diarization Challenge 2018," in *Interspeech*, 2018.

[10] X. Qin, C. Wang, Y. Ma, M. Liu, S. Zhang, and M. Li, "Our Learned Lessons from Cross-Lingual Speaker Verification: The CRMI-DKU System Description for the Short-Duration Speaker Verification Challenge 2021," in *Proc. Interspeech 2021*, 2021, pp. 2317–2321.

[11] A. Zhang, Q. Wang, Z. Zhu, J. W. Paisley, and C. Wang, "Fully supervised speaker diarization," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6301–6305, 2018.

[12] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with permutation-free objectives," in *Interspeech*, 2019.

[13] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with self-attention," *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 296–303, 2019.

[14] S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, and K. Nagamatsu, "End-to-End Speaker Diarization for an Unknown Number of Speakers with Encoder-Decoder Based Attractors," in *Proc. Interspeech 2020*, 2020, pp. 269–273.

[15] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 241–245.

[16] M. Kolbæk, D. Yu, Z.-H. Tan, and J. Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1901–1913, 2017.

[17] K. Kinoshita, M. Delcroix, and N. Tawara, "Integrating end-to-end neural and clustering-based diarization: Getting the best of both worlds," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 7198–7202.

[18] K. Kinoshita, M. Delcroix, and T. Iwata, "Tight integration of neural- and clustering-based diarization through deep unfolding of infinite gaussian mixture model," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 8382–8386.

[19] K. Kinoshita, M. Delcroix, and N. Tawara, "Advances in Integration of End-to-End Neural and Clustering-Based Diarization for Real Conversational Speech," in *Proc. Interspeech 2021*, 2021, pp. 3565–3569.

[20] J. Su, M. Zhu, A. Murtadha, S. Pan, B. Wen, and Y. Liu, "ZLPR: A Novel Loss for Multi-label Classification," *ArXiv*, vol. abs/2208.02955, 2022.

[21] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, and Y. Wei, "Circle loss: A unified perspective of pair similarity optimization," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6397–6406, 2020.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[23] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[24] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, arXiv:1510.08484v1.

[25] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *IEEE International Conference on Acoustics*, 2017.