



Pushing the Limits of End-to-End Diarization

Samuel J. Broughton, Lahiru Samarakoon

Fano, Hong Kong SAR, China

{samuel.broughton, lahiru}@fano.ai

Abstract

In this paper, we present state-of-the-art diarization error rates (DERs) on multiple publicly available datasets, including AliMeeting-far, AliMeeting-near, AMI-Mix, AMI-SDM, DIHARD III, and MagicData RAMC. Leveraging EEND-TA, a single unified non-autoregressive model for end-to-end speaker diarization, we achieve new benchmark results, most notably a DER of 14.49% on DIHARD III. Our approach scales pre-training through 8-speaker simulation mixtures, ensuring each generated speaker mixture configuration is sufficiently represented. These experiments highlight that EEND-based architectures possess a greater capacity for learning than previously explored, surpassing many existing diarization solutions while maintaining efficient speeds during inference.

Index Terms: EEND, EEND-TA, Speaker Diarization

1. Introduction

The perfect speaker diarization system should partition any given input audio signal into distinct segments, each correctly labeled with the corresponding speaker identity. In practice, this task is notoriously difficult due to the variability of input recordings. A robust system must handle varying degrees of background noise, a wide dynamic range, multiple speakers with diverse accents and languages, overlapping speech, and a broad range of audio quality captured from many different recording devices and conditions.

Traditional diarization systems are typically comprised of several components that are trained separately on non-diarization objectives [1]. Such cascaded pipelines usually include voice activity detection, speaker embedding extraction and clustering modules. Here, speaker embeddings are extracted from voice active frames so that a clustering algorithm can group together segments belonging to the same speaker. Generally, these solutions cannot handle overlapped regions of speech their reliance on multiple components increases both complexity and inference time.

End-to-end neural approaches to diarization directly address overlapping speech. End-to-end Neural Diarization (EEND) formulated diarization as a multi-label classification problem, outputting speaker labels for a fixed number of speakers [2, 3]. Many studies since enhanced EEND by adding an an Encoder Decoder Attractor (EDA) based calculation module to handle a variable number of speakers [4, 5], and others have improved EDA with attention-based attractor mechanisms [6–11].

However, end-to-end neural networks such as EEND require a substantial amount of data to learn from, and annotated diarization datasets are scarce and expensive to label. To address this, it has become standard practice to generate simulated training mixtures with multiple speakers and pre-train

models before fine-tuning on a real-world dataset. Some works have further investigated the original simulation strategy proposed to construct the audio mixtures, developing an algorithm with the goal of better emulating aspects of natural conversations [12–14]. This technique has been shown to increase diarization performance on telephony style data with little improvement on wide-band data [9].

To date, few studies have examined the effects of simulating mixtures with a high number of speakers. One study used 2,500 hours of simulated audio with up to 10 speakers [10], while another pre-trained on 16,700 hours with up to 18 speakers [15]. However, the impact of simulating a higher number of speakers remains largely unexplored.

In this work, we combine recent EEND-based advances with scaled pre-training, simulating mixtures for up to 8 speakers and increasing the total pre-training set to over 80,000 hours. This approach achieves state-of-the-art Diarization Error Rate (DER) for both end-to-end and cascaded systems across multiple datasets: AliMeeting-far (11.41%), AliMeeting-near (8.55%), AMI-Mix (11.04%), AMI-SDM (15.16%), DIHARD III (14.49%) and MagicData RAMC (10.43%).

2. Related Work

EEND-based methods have become the foundation of many diarization solutions. A common approach is to use an EEND module on short sliding windows to obtain local diarization results, then extract speaker embeddings and apply a clustering algorithm for global segmentation [16–18].

To better address end-to-end use cases, EEND was extended with an EDA network, which outperforms conventional EEND and, in theory, handles an unlimited number of speakers. However, diarization performance degrades when the model encounters more speakers than seen during training [4, 5].

Subsequent works have improved both the backbone and prediction head of EEND-EDA. Performance gains were demonstrated by replacing Transformers with Conformer layers [19, 20], introducing Transformer-based attractor calculations [6–11], and using deep supervision training objectives [7, 10, 11]. Other approaches introduced supplementary inputs for further refinement, such as summary representations [21], or diarization outputs for attractor estimation [6].

Because end-to-end models require large amounts of training data, it is common practice to pre-train models on simulated mixtures [2, 13, 14]. Increasing the number of simulated speakers within an audio mixture during pre-training can boost performance at fine-tuning [9, 22].

3. Method

3.1. Model Architecture: EEND-TA

For this work, we selected EEND-TA as the network model architecture, a single unified non-autoregressive diarization model. We chose EEND-TA for its efficient design, which omits additional techniques such as iterative refinement [6], or self-conditioning [7]. Previously, EEND-TA had only ever been shown to handle diarization for up to 4 speakers. The model consists of a Conformer encoder [23], a combiner, and a Transformer decoder. Consistent with the original work, we use a modified Conformer encoder to accommodate the Conversational Summary Vector (CSV) [21]. This is a learnable special token that is prepended to the feature sequence before passing through the Conformer encoder. CSV was shown to improve performance on recordings with a higher number of speakers. The combiner joins the CSV with a set of learnable global embeddings, which are then input to the Transformer decoder. We use a standard Transformer decoder without positional embeddings to generate candidate speaker-wise attractors. Speaker existence posterior probabilities are then computed using a linear layer followed by a Sigmoid function. Final diarization outputs are computed by a Sigmoid function acting on the matrix product of the encoder output embeddings and speaker-wise attractors.

3.2. Scaled Pre-Training

We scaled pre-training by increasing the maximum possible number of speakers within a simulated mixture to 8. Simulated mixtures are generated by using a modified version of the standard simulated mixture algorithm [2, Alg. 1].

Appropriate average silence interval values must be selected for each set of simulated mixtures, ranging from 1 to 8 speakers. Consistent with many other studies, for mixtures containing 1, 2, 3 or 4 speakers we applied the standard average silence interval values of $\beta = 2, 2, 5, 9$, respectively. To determine β values for mixtures containing 5, 6, 7 or 8 speakers, we calculated the average silence duration for each speaker in our real diarization dataset. We then took the mean silence duration for all speakers within each dataset split, where each dataset split represents recordings containing 5, 6, 7 or 8 speakers. This resulted in $\beta = 34, 54, 47, 50$ for mixtures with 5, 6, 7 or 8 speakers, respectively.

However, using such large β values can lead to simulated mixtures with extended periods of silence where no speaker is active. To address this, we modified the original algorithm to simply replace all silences longer than 5 seconds with a randomly chosen silence duration between 1 and 5 seconds.

4. Experiments

4.1. Data

Pre-training mixtures are generated from the LibriSpeech Corpus [29], using the modified simulation strategy described in Section 3.2. Table 3 compares the real datasets used at fine-tuning with the simulated mixtures used for pre-training. In total, we generated 100,000 simulated recordings for each possible number of active speakers in a mixture, resulting to 800,000 mixtures with 1 to 8 speakers. This amounted to over 80,000 hours of simulated data. There are two models shown in Tables 1 and 2 that use a total of 400,000 mixtures, “EEND-TA C4 (400 k) S4” and “EEND-TA C4 (400 k) S8”. The “S4” model is pre-trained with 1 to 4 speakers containing 100,000 mixtures

per speaker, a setup more similar to the original work, whereas, the “S8” model is pre-trained with 1 to 8 speakers, using 50,000 simulated recordings per speaker mixture.

For model fine-tuning, we use the publicly available datasets: AISHELL-4 [30], AliMeeting [31], AMI-Mix (headset mixtures) and AMI-SDM (single distant microphone) [32], CALLHOME [33], DIHARD III, MagicData RAMC [34] and VoxConverse (version 0.3) [35]. Where available, we use the official training, validation and testing splits provided by the dataset. For DIHARD III and VoxConverse we use the evaluation data for testing and split the train set into training and validation subsets using a 80%:20% split. The AMI-Mix and AliMeeting datasets containing multi-channel audio recordings are downmixed to a single channel. We refer to the AliMeeting far-field 8 microphone array recordings and headset microphone recordings as “AliMeeting-far” and “AliMeeting-near”, respectively.

4.2. Experimental Setup

The input to EEND-TA is 23 log-dimensional Mel-filterbank features, extracted using a window length of 25ms and hop size of 10ms. The input feature sequence is first downsampled by a factor of 10 using convolutional layers with kernel sizes {3, 5}, strides {2, 5} and output feature dimension size 256. The downsampled sequence is then passed to a Conformer encoder consisting of 6 stacked layers each with 4 attention heads and feed-forward layers with 1024 hidden units. Before passing through the encoder, the input feature sequence is concatenated to a randomly initialized CSV representation. Models denoted as “EEND-TA C4” in Tables 1 and 2 use a Conformer encoder consisting of only 4 stacked layers.

The Transformer Attractor calculation module consists of 3 Transformer decoder layers with 4 attention heads and a feed-forward dimension size of 1024. TA takes $S + 1$ randomly initialized 256 dimensional input queries, each of which are combined with CSV. Our models were trained to predict up to $S = 8$ speakers. Only the “EEND-TA C4 (400 k) S4” model was trained to predict up to $S = 4$ speakers.

The original EDA module was used to train “EEND-EDA[§] (ours)” seen in Tables 1 and 2. This model was trained to predict up to $S = 8$ speakers and also makes use of the CSV [21].

Each Model was pre-trained for 2.5 M steps and fine-tuned for 250 k steps, all in bfloat16 precision. Pre-training used a batch size of 256 (32×8) mixtures per step that were randomly cropped to an utterance length of 220 seconds. Fine-tuning used a batch size of 8 recordings cropped to 600 seconds. Chunk shuffling was applied only at fine-tuning [20]. The Adam optimizer and Noam scheduler with 100 k warm-up steps was used for pre-training, and the Adam optimizer with a fixed learning rate of 1×10^{-5} was used for fine-tuning.

Model weights of the last pre-training checkpoint are used to initialize each model for fine-tuning. For “EEND-EDA[§] (ours)” and “EEND-TA[§]”, we load only the pre-trained Conformer encoder from EEND-TA and randomly initialize the weights for the EDA and TA head modules before fine-tuning. The top 10 best checkpoints in terms of validation DER performance are averaged to create the model weights for inference. We conducted 20 k steps of dataset-specific fine-tuning after the aggregated fine-tuning stage of 250 k steps, marked as “EEND + FT” in Table 1. All models are evaluated using an attractor existence threshold of 0.5 and diarization threshold of 0.5.

Table 1: *Diarization Error Rate (DER) across several datasets. Lower is better. Updated state-of-the-art results are marked in bold. Values in parenthesis are calculated with a 0.25 second forgiveness collar. “+FT” denotes results for fine-tuning on a single dataset.*

| Model | Dataset | | | | | | | | |
|---------------------------------------|-----------|----------------|-----------------|--------------|--------------|----------------|--------------|----------------|---------------------------|
| | AISHELL-4 | AliMeeting-far | AliMeeting-near | AMI-Mix | AMI-SDM | CALLHOME | DIHARD III | MagicData RAMC | VoxConverse |
| VAD+VBx+OSD [10] | 15.84 | 28.84 | 22.59 | 22.42 | 34.61 | 26.18 | 20.28 | 18.19 | (6.12) |
| EEND [3, 5] | — | — | — | 27.70 | — | (21.19) | 22.64 | — | — |
| EEND-EDA [5] | — | — | — | 15.80 | — | (12.88) | 20.69 | — | — |
| EEND-EDA [§] (ours) | 13.43 | 12.30 | 9.05 | 11.31 | 16.29 | 17.62 | 15.02 | 10.60 | 15.98 |
| DiaPer [10] | 29.0 | 20.7 | 17.8 | 23.9 | 40.7 | 24.16 | 21.1 | 16.1 | (19.1) |
| pyannote.audio v3.1 [†] [24] | 12.2 | 24.4** | — | 18.8 | 22.4 | 28.4 | 21.7 | 22.2 | 11.3 |
| pyannoteAI [†] | 11.9 | 22.5** | — | 16.6 | 20.9 | 22.2 | 17.2 | 18.4 | 9.4 |
| EEND-M2F [11] | 15.56 | 13.20 | 10.77 | 13.86 | 19.83 | 21.28 | 16.28 | 11.13 | 15.99 |
| +0.25s collar | (10.75) | (5.87) | (5.20) | (9.16) | (14.29) | (14.87) | (8.93) | (6.52) | (12.02) |
| EEND-M2F + FT [11] | 13.98 | 13.40 | 10.45 | 12.62 | 18.85 | 23.44 | 16.07 | 11.09 | 16.28 |
| +0.25s collar | (9.34) | (6.11) | (5.02) | (7.92) | (13.33) | (16.72) | (8.82) | (6.46) | (12.36) |
| EEND-TA C4 (400 k) S4 | 25.68 | 12.68 | 10.36 | 12.88 | 20.08 | 19.24 | 17.50 | 10.97 | 21.75 |
| EEND-TA C4 (400 k) S8 | 22.64 | 13.23 | 9.92 | 15.27 | 19.16 | 18.91 | 15.71 | 10.43 | 17.06 |
| EEND-TA C4 | 15.41 | 11.73 | 9.29 | 14.56 | 17.68 | 19.21 | 15.13 | 10.30 | 17.03 |
| EEND-TA [§] | 15.09 | 11.45 | 9.05 | 11.31 | 16.13 | 17.51 | 15.29 | 10.57 | 15.75 |
| EEND-TA | 15.31 | 12.65 | 8.60 | 11.06 | 15.16 | 16.91 | 14.76 | 10.43 | 15.44 |
| +0.25s collar | (10.82) | (6.83) | (4.09) | (7.19) | (10.19) | (10.97) | (8.16) | (5.79) | (11.57) |
| EEND-TA + FT | 12.21 | 11.41 | 8.55 | 11.04 | 15.33 | 17.24 | 14.49 | 10.55 | 14.29 |
| +0.25s collar | (7.54) | (5.16) | (4.07) | (7.15) | (10.27) | (11.09) | (8.11) | (5.96) | (10.41) |
| State-of-the-art (as of Feb. 2025) | 11.7 | 13.20 | 10.45 | 12.62 | 15.4 | (10.08) | 16.07 | 11.09 | (4.0) [*] |
| Source | [25] | [11] | [11] | [11] | [25] | [26] | [27] | [8] | [28] |

Some results from other works may use differing acoustic setups.

[†] Results for pyannote systems are taken directly from

<https://github.com/pyannote/pyannote-audio/blob/develop/README.md>, as of commit 93ad8b9.

[§] Only pre-trained EEND encoder weights loaded for fine-tuning, model head parameters are randomly initialized.

^{*} Potentially biased, as model was tuned and validated on VoxConverse test set.

Table 2: *Diarization Error Rate (DER) and Mean Speaker Counting Error (MSCE) per speaker on all datasets. Lower is better.*

| Model | 1-spkr | | 2-spkr | | 3-spkr | | 4-spkr | | 5-spkr | | 6-spkr | | 7-spkr | | 8-spkr | |
|------------------------------|--------|------|--------|------|--------|------|--------|------|--------|------|--------|------|--------|------|--------|------|
| EEND-TA C4 (400 k) S4 | 7.68 | 0.20 | 9.36 | 0.03 | 16.37 | 0.45 | 17.02 | 0.44 | 30.55 | 1.43 | 23.02 | 2.23 | 24.15 | 3.28 | 27.31 | 4.19 |
| EEND-TA C4 (400 k) S8 | 7.80 | 0.33 | 9.19 | 0.06 | 15.97 | 0.45 | 17.07 | 0.54 | 27.10 | 1.00 | 20.25 | 1.28 | 19.12 | 1.74 | 21.64 | 2.11 |
| EEND-TA C4 | 8.21 | 0.47 | 9.60 | 0.06 | 15.07 | 0.40 | 14.93 | 0.45 | 21.27 | 0.86 | 16.46 | 0.97 | 16.62 | 1.52 | 23.55 | 1.30 |
| EEND-EDA [§] (ours) | 7.47 | 0.23 | 9.01 | 0.04 | 14.72 | 0.37 | 14.71 | 0.42 | 20.06 | 0.89 | 16.69 | 1.10 | 15.84 | 1.61 | 20.01 | 2.22 |
| EEND-TA [§] | 7.14 | 0.23 | 8.97 | 0.04 | 14.35 | 0.41 | 14.55 | 0.38 | 21.56 | 0.89 | 16.78 | 1.03 | 16.27 | 1.65 | 20.55 | 2.07 |
| EEND-TA | 8.86 | 0.27 | 9.02 | 0.07 | 15.02 | 0.40 | 14.02 | 0.34 | 23.81 | 1.00 | 14.55 | 0.97 | 13.47 | 1.22 | 18.96 | 1.56 |

[§] Only pre-trained EEND encoder weights loaded for fine-tuning, model head parameters are randomly initialized.

5. Discussion

5.1. Main Results

The main results for our models, and current state-of-the-art results as of February 2025, are presented in Table 1. Unless stated otherwise, we report the least forgiving diarization error rates (DERs) by scoring all speech including overlaps, with no oracle speaker counting, no oracle voice activity detection,

no hyper-parameter tuning on specific datasets and no forgiveness collar. It is important to note that each state-of-the-art result shown is obtained by fine-tuning a model on that specific dataset, except for AliMeeting-far, AMI-Mix and AMI-SDM.

Without dataset-specific fine-tuning, our model achieves state-of-the-art performance on 6 out of the 9 test sets: AliMeeting-far, AliMeeting-near, AMI-Mix, AMI-SDM, DIHARD III and MagicData RAMC. Impressively, EEND-TA im-

Table 3: Train Dataset Statistics for 1 to 8 Speakers. “Seen (%)” refers to the percentage of active speakers per utterance seen by the model across an entire epoch.

| # Spks | Overlap (%) | | Duration (hrs) | | Seen (%) | |
|--------|-------------|------|----------------|----------|----------|------|
| | Real | Sim. | Real | Sim. | Real | Sim. |
| 1 | N/A | N/A | 2.7 | 9047.1 | 1.6 | 12.5 |
| 2 | 5.0 | 39.5 | 246.2 | 10,050.5 | 41.4 | 12.5 |
| 3 | 16.0 | 38.0 | 35.3 | 12,757.3 | 8.4 | 12.5 |
| 4 | 25.0 | 33.9 | 321.1 | 12,377.6 | 36.3 | 12.5 |
| 5 | 10.1 | 13.7 | 46.1 | 10,130.5 | 5.7 | 12.5 |
| 6 | 8.2 | 10.7 | 4.1 | 9821.4 | 1.7 | 12.5 |
| 7 | 9.0 | 14.3 | 17.0 | 10,481.9 | 2.5 | 12.6 |
| 8 | 15.0 | 15.6 | 2.1 | 11,083.9 | 2.2 | 12.4 |

Table 4: Real Time Factor (RTF) was computed based on the total time to decode the DIHARD III evaluation set, one by one.

| Model | # Params (M) | RTF | |
|--------------------------|------------------|----------------------|----------------------|
| | | CPU | GPU |
| pyannote.audio v3.1 [24] | 8.1 [‡] | 3.5×10^{-1} | 1.1×10^{-2} |
| EEND-EDA (ours) | 11.3 | 2.7×10^{-3} | 7.1×10^{-4} |
| EEND-M2F [11] | 16.3 | 3.6×10^{-3} | 2.5×10^{-4} |
| EEND-TA | 13.3 | 2.2×10^{-3} | 2.3×10^{-4} |

[‡] Total parameters for the segmentation and embedding model

proves the DER for DIHARD III by a relative 8.15%. After dataset-specific fine-tuning (EEND-TA + FT) we further improve upon many DER results for a number of test sets.

There are three datasets where EEND-TA, with or without dataset-specific fine-tuning, does not beat current state-of-the-art results: AISHELL-4, CALLHOME and VoxConverse. Contrary to EEND-M2F, EEND-TA reaches near state-of-the-art performance on CALLHOME, while its results on AISHELL-4 and VoxConverse (containing 5 to 8 speakers and up to 21 speakers, respectively) are still comparable to EEND-M2F. Most of the errors made on these datasets stem from speaker confusion. After fine-tuning solely on AISHELL-4, our model achieves near state-of-the-art results.

5.2. Scaling Pre-training

By scaling model pre-training we show that EEND-TA had more capacity to learn than originally presented [8]. Table 2 shows the progression of EEND-TA models before and after scaled pre-training per speaker. Looking at the “S4” and “S8” models, as expected, by including 5 to 8 speaker pre-training and allowing EEND-TA to predict up to 8 speakers, we see an improvement for both DER and MSCE for 5 to 8 speaker recordings with no major changes to results for 1 to 4 speakers. Doubling the pre-training data to 100,000 mixtures per speaker, “EEND-TA C4” further improves upon results, especially for recordings containing 3 to 7 speakers. Additionally, this model achieves the lowest DER on the MagicData RAMC test set.

An increase to 6 Conformer encoder layers also demonstrates a greater capacity to learn. Particularly for recordings containing 6 to 8 speakers, there are relatively large reductions to DER. Table 2 also shows similar performance between EDA and TA variants. By comparing models that were both trained with randomly initialized heads, EEND-EDA shows consistent improvement for 5 to 8 speaker recordings. These results are competitive to models that fully load pre-trained model weights.

An important point to highlight is that even though our pre-training strategy crops each item in a batch to 220s, Table 3 shows that for each epoch during pre-training the model sees sequences of 220s containing 8 speakers 12.4% of the time.

Overall, the pre-training dataset has doubled in size, as has the maximum possible number of speakers within a single simulated mixture. This allowed for an increase to EEND-TAs total number of parameters by 30% (10.2M to 13.3M). However, when compared to other speech domains, these models are still relatively small and lightweight. Given a much larger and more diverse simulated pre-training dataset, we hypothesize that diarization models can grow in size to match their ASR counterparts. Even though the pre-training dataset used here is approximately 80,000 hours in length, this can just be viewed as an 80-fold augmentation of the original 960 hour long train set of the LibriSpeech Corpus. Therefore, future work will look towards further scaling the model size, pre-training dataset size, and pre-training dataset diversity.

5.3. Real Time Factor

When choosing a model for production, it is important to assess speed. Table 4 shows the Real Time Factor (RTF) when running inference on the DIHARD III evaluation set for best performing end-to-end diarization models and pyannote.audio v3.1. To calculate the RTF, we load the respective system and each recording of the 33 hour long evaluation set into memory. Audios are loaded as PyTorch Tensors. For the GPU benchmark, the diarization model or pipeline is pre-loaded into the GPU’s vRAM. We then measure the total amount of time it takes each system to decode the entire list of Tensors, one by one. Both CPU and GPU benchmarks were conducted by using a single Intel Xeon Gold 6330 Processor. The GPU benchmark used a single NVIDIA RTX A6000.

As expected, end-to-end models are much faster at diarizing an audio when compared to clustering-based methods. EEND-TA is faster than EEND-M2F at both CPU and GPU decoding due to reduced model size and complexity. Here, EEND-TA processes the DIHARD III evaluation set around 460 times faster than real time with CPU decoding and 4290 times faster with GPU decoding, compared to pyannote.audio v3.1’s more cascaded approach, which is only 3 and 92 times faster with CPU and GPU decoding, respectively. EEND-EDA is slower than EEND-TA due to the sequential nature of LSTMs used in the EDA module. Given pre-processed log Mel-filterbank features, EEND-TA can process the entire combined 158 hours long test set in 97s on GPU, 5870 times faster than real time with GPU VRAM reaching a maximum of only 1.6 GiB.

6. Conclusion

In this paper, we demonstrated that end-to-end diarization models still possess a significant capacity for learning. By scaling model pre-training with EEND-TA, we achieved state-of-the-art results on AliMeeting-far, AliMeeting-near, AMI-Mix, AMI-SDM, DIHARD III, and MagicData RAMC. We also provided a strategy for generating up to 8-speaker simulation mixtures, so that a 220s random crop used for pre-training is highly likely to include all active speakers. Furthermore, our findings show that end-to-end methods yield substantially lower real-time factors, making them well suited to production environments. To the best of our knowledge, as of February 2025, the results presented in this paper outperform all other end-to-end diarization models.

7. References

- [1] 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. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0885230821001121>
- [2] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-End Neural Speaker Diarization with Permutation-Free Objectives," in *Proc. Interspeech 2019*, 2019, pp. 4300–4304.
- [3] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with self-attention," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 296–303.
- [4] 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.
- [5] S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, and P. García, "Encoder-decoder based attractors for end-to-end neural diarization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1493–1507, 2022.
- [6] M. Rybicka, J. Villalba, N. Dehak, and K. Kowalczyk, "End-to-end neural speaker diarization with an iterative refinement of non-autoregressive attention-based attractors," in *Proc. Interspeech*, vol. 2022, 2022, pp. 5090–5094.
- [7] Y. Fujita, T. Komatsu, R. Scheibler, Y. Kida, and T. Ogawa, "Neural diarization with non-autoregressive intermediate attractors," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [8] L. Samarakoon, S. J. Broughton, M. Härkönen, and I. Fung, "Transformer attractors for robust and efficient end-to-end neural diarization," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.
- [9] Z. Chen, B. Han, S. Wang, and Y. Qian, "Attention-based encoder-decoder end-to-end neural diarization with embedding enhancer," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 1636–1649, 2024.
- [10] F. Landini, T. Stafylakis, L. Burget *et al.*, "Diaper: End-to-end neural diarization with perceiver-based attractors," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.
- [11] M. Härkönen, S. J. Broughton, and L. Samarakoon, "Eend-m2f: Masked-attention mask transformers for speaker diarization," in *Proc. Interspeech 2024*, 2024, pp. 37–41.
- [12] N. Yamashita, S. Horiguchi, and T. Homma, "Improving the naturalness of simulated conversations for end-to-end neural diarization," in *Odyssey*, 2022, pp. 133–140.
- [13] F. Landini, A. Lozano-Diez, M. Diez, and L. Burget, "From simulated mixtures to simulated conversations as training data for end-to-end neural diarization," in *Proc. Interspeech 2022*, 2022, pp. 5095–5099.
- [14] F. Landini, M. Diez, A. Lozano-Diez, and L. Burget, "Multi-speaker and wide-band simulated conversations as training data for end-to-end neural diarization," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [15] Z. Pan, G. Wichern, F. G. Germain, A. Subramanian, and J. Le Roux, "Late audio-visual fusion for in-the-wild speaker diarization," in *2024 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*. IEEE, 2024, pp. 174–178.
- [16] 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.
- [17] H. Bredin and A. Laurent, "End-to-end speaker segmentation for overlap-aware resegmentation," in *Proc. Interspeech 2021*, 2021, pp. 3111–3115.
- [18] H. Bredin, "pyannote.audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe," in *Proc. INTERSPEECH 2023*, 2023, pp. 1983–1987.
- [19] Y. C. Liu, E. Han, C. Lee, and A. Stolcke, "End-to-End Neural Diarization: From Transformer to Conformer," in *Proc. Interspeech 2021*, 2021, pp. 3081–3085.
- [20] T.-Y. Leung and L. Samarakoon, "Robust End-to-End Speaker Diarization with Conformer and Additive Margin Penalty," in *Proc. Interspeech 2021*, 2021, pp. 3575–3579.
- [21] S. J. Broughton and L. Samarakoon, "Improving End-to-End Neural Diarization Using Conversational Summary Representations," in *Proc. INTERSPEECH 2023*, 2023, pp. 3157–3161.
- [22] S. Horiguchi, N. Yalta, P. Garcia, Y. Takashima, Y. Xue, D. Raj, Z. Huang, Y. Fujita, S. Watanabe, and S. Khudanpur, "The hitachi-ju dihard iii system: Competitive end-to-end neural diarization and x-vector clustering systems combined by dover-lap," *arXiv preprint arXiv:2102.01363*, 2021.
- [23] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 5036–5040.
- [24] A. Plaquet and H. Bredin, "Powerset multi-class cross entropy loss for neural speaker diarization," in *Proc. INTERSPEECH 2023*, 2023, pp. 3222–3226.
- [25] J. Han, F. Landini, J. Rohdin, A. Silnova, M. Diez, and L. Burget, "Leveraging self-supervised learning for speaker diarization," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.
- [26] Z. Chen, B. Han, S. Wang, and Y. Qian, "Attention-based encoder-decoder end-to-end neural diarization with embedding enhancer," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 1636–1649, 2024.
- [27] M.-K. He, J. Du, Q.-F. Liu, and C.-H. Lee, "Ansd-ma-mse: Adaptive neural speaker diarization using memory-aware multi-speaker embedding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1561–1573, 2023.
- [28] S. Baroudi, H. Bredin, A. Plaquet, and T. Pellegrini, "pyannote.audio speaker diarization pipeline at voxsrc 2023."
- [29] 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.
- [30] Y. Fu, L. Cheng, S. Lv, Y. Jv, Y. Kong, Z. Chen, Y. Hu, L. Xie, J. Wu, H. Bu, X. Xu, J. Du, and J. Chen, "AISHELL-4: An Open Source Dataset for Speech Enhancement, Separation, Recognition and Speaker Diarization in Conference Scenario," in *Proc. Interspeech 2021*, 2021, pp. 3665–3669.
- [31] F. Yu, S. Zhang, Y. Fu, L. Xie, S. Zheng, Z. Du, W. Huang, P. Guo, Z. Yan, B. Ma, X. Xu, and H. Bu, "M2MeT: The ICASSP 2022 multi-channel multi-party meeting transcription challenge," in *Proc. ICASSP*. IEEE, 2022.
- [32] W. Kraaij, T. Hain, M. Lincoln, and W. Post, "The ami meeting corpus," in *Proc. International Conference on Methods and Techniques in Behavioral Research*, 2005.
- [33] NIST, "The 2000 NIST speaker recognition evaluation plan." Tech. Rep., 2009.
- [34] Z. Yang, Y. Chen, L. Luo, R. Yang, L. Ye, G. Cheng, J. Xu, Y. Jin, Q. Zhang, P. Zhang, L. Xie, and Y. Yan, "Open Source MagicData-RAMC: A Rich Annotated Mandarin Conversational(RAMC) Speech Dataset," in *Proc. Interspeech 2022*, 2022, pp. 1736–1740.
- [35] J. S. Chung, J. Huh, A. Nagrani, T. Afouras, and A. Zisserman, "Spot the Conversation: Speaker Diarisation in the Wild," in *Proc. Interspeech 2020*, 2020, pp. 299–303.