# Joint Training of Speaker Embedding Extractor, Speech and Overlap Detection for Diarization

Petr Pálka[1], Federico Landini[1], Dominik Klement[1], Mireia Diez[1], Anna Silnova[1], Marc Delcroix[2], Lukáš Burget[1]
[1]Brno University of Technology, Speech@FIT, Czechia [2]NTT Corporation, Japan

***Abstract*—In spite of the popularity of end-to-end diarization systems nowadays, modular systems comprised of voice activity detection (VAD), speaker embedding extraction plus clustering, and overlapped speech detection (OSD) plus handling still attain competitive performance in many conditions. However, one of the main drawbacks of modular systems is the need to run (and train) different modules independently. In this work, we propose an approach to jointly train a model to produce speaker embeddings, VAD and OSD simultaneously and reach competitive performance at a fraction of the inference time of a modular approach. Furthermore, the joint inference leads to a simplified overall pipeline which brings us one step closer to a unified clustering-based method that can be trained end-to-end towards a diarization-specific objective.**

***Index Terms*—speaker diarization, speaker embedding, voice activity detection, overlapped speech detection**

## I. Introduction

Until a few years ago, competitive speaker diarization systems were mostly modular [1]–[3], i.e., consisting of different modules to handle voice/speech activity detection (VAD/SAD), embedding extraction over uniform segmentation, clustering, optional resegmentation, and overlapped speech detection (OSD) and handling. However, end-to-end models such as end-to-end neural diarization (EEND) [4], [5], and two-stage systems such as target-speaker voice activity detection (TS-VAD) [6] or end-to-end with vector clustering (E2E-VC) [7]–[10] have recently gained more and more attention. The reasons for this are mainly their inherent ability to handle overlapped speech (where modular systems underperform) and fewer steps at inference time. Nevertheless, in contrast with modular systems, single-stage end-to-end systems do not handle well scenarios with many speakers [11] and they are very data-hungry, requiring high volumes of training data with diarization annotations.

While two-stage systems produce per-frame speaker labels directly with a neural network (NN), they still build on clustering of embeddings: TS-VAD normally uses a clustering-based approach for initialization and recent competitive approaches based on E2E-VC [12] make use of the best speaker embedding extractors available together with clustering to reconcile short-segment decisions. Besides, modular systems can still attain competitive performance in certain scenarios [13], so speaker embedding extraction and clustering are still very relevant for diarization today.

Computing on IT4I supercomputer was supported by the Czech Ministry of Education, Youth and Sports through the e-INFRA CZ (ID:90254).

Speaker embedding extraction and clustering have been the main components of modular systems for more than a decade. Since the development of x-vectors [14], the embeddings have been NN-based with new versions such as ResNet [15], [16], ECAPA-TDNN [17] or ECAPA2 [18] providing better and better results for speaker recognition and verification. These models are trained and utilized on at least a few seconds-long recordings for these tasks, but for diarization, embeddings are extracted on shorter segments since speakers can have short turns. However, the models are not designed for such usage, and tailoring them could lead to better performance [19]–[23].

In the context of clustering-based diarization, VAD (and optionally OSD) is needed. In this work, we modify the embedding extractor to produce per-frame embeddings for the whole recording at once, naturally avoiding multiple calls to the embedding extraction routine and speeding up the process, while producing VAD and OSD labels for each embedding. This is done by removing the pooling mechanism and introducing linear layers to produce VAD and OSD decisions from the embeddings. Moreover, we train the model for VAD+OSD and for embedding extraction on different data, thus taking advantage of different types of supervision that different corpora might offer, without generating synthetic training data, which is usual for E2E systems [4], [24]–[26].

In related works like [27], before x-vectors became popular, a single NN was used to extract per-frame speaker embeddings and produce VAD and OSD labels. However, the quality of the embeddings was restricted by the contrastive loss used to train the model and the limited speaker set contained in diarization-annotated datasets. In [28], per-frame embeddings were produced in a teacher-student framework where the teacher model produced per-segment embeddings. More recently, in [29], [30], the embedding extractor was used to provide VAD labels as a by-product of using information encoded in intermediate representations in a weakly supervised VAD framework.

The results obtained with our proposed approach show that it is possible to train a single model for the three tasks (VAD, embedding extraction, OSD) and produce high-quality embeddings at a higher frequency. This opens up the space for building speaker verification systems that can discard silence- and overlap-related frames before producing per-utterance embeddings. Moreover, the results are encouraging in our plan to combine this model with discriminative VBx (DVBx) [31], which will enable training of the whole modular pipeline in an end-to-end fashion towards a diarization-specific objective.

## II. Standard Methods

### A. Diarization system pipeline

We follow a standard modular pipeline (Figure 1a): VAD, embedding extraction, and clustering of those embeddings. Since clustering-based approaches assume a single speaker for each embedding, in order to handle overlaps between speakers, OSD is necessary. We assign second speakers in overlap segments based on the heuristic [32] that assigns the second closest (in terms of time) speaker. The embeddings, VAD and OSD labels can be produced by specific models trained for each of the tasks or, as we present in this work, produced by the joint speaker embedding extractor, VAD, and OSD model. The proposed pipeline is shown in Figure 1b. We use VBx [33] for clustering the embeddings with DVBx [31] to find suitable hyperparameters.

### B. Baseline embedding extraction

A typical embedding extractor used in speaker recognition (Figure 2a) consists of an encoder processing the information frame-by-frame, a pooling mechanism, and a feed-forward NN processing segment-level information. In all of our experiments, we use a ResNet-101 architecture [15] as an encoder for the embedding extractor [16]. However, the same ideas and extensions can be applied to other common architectures. The encoder transforms a sequence of 64-dimensional log Mel-filterbanks extracted every 10 ms into a shorter sequence (one vector per 80 ms of the original audio) of internal 8192-dimensional representations. Note that the theoretical receptive field of ResNet101 is slightly longer than 2.5 s, so each of the internal representations is estimated on a relatively long segment of speech, much more than only 80 ms.

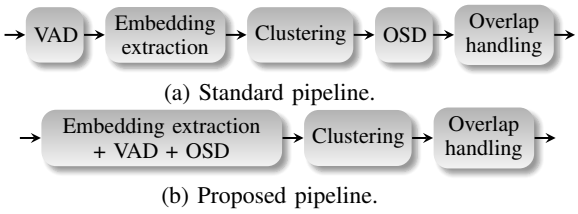
The encoder is followed by a pooling layer that combines the information along the temporal dimension. Thus, the whole audio is represented by a single fixed-dimensional vector independently of its length, thus "per-segment" embedding. As a pooling layer, we chose the commonly used temporal statistical pooling [14] - the concatenation of the means and standard deviations of temporal representations. Then, the pooled vector is passed through a fully connected layer, which reduces its dimensionality, and finally enters the classification head during training. At inference time, the activations of the fully connected layer are used as speaker embeddings.

For the sake of making fair comparisons, all systems utilize a ResNet-101 trained with the WeSpeaker toolkit [34]. The models are trained to classify speakers in the training set with AAM loss [35], [36] on VoxCeleb2 (VC2) [37]. Training hyperparameters such as learning rate, margin scheduler, number of epochs, etc., follow the official WeSpeaker VoxCeleb recipe.

## III. Proposed Modifications

### A. High-resolution speaker embedding extraction

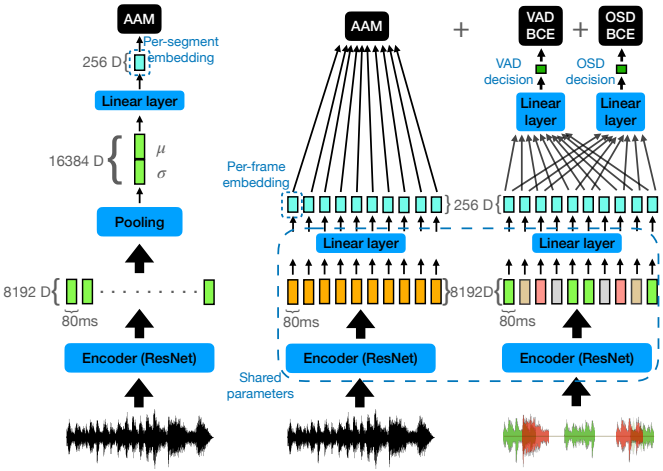
The first modification to the usual architecture is the removal of the pooling layer common to all embedding extractors. We add a linear layer to reduce the dimensionality of the embeddings coming from the ResNet from 8192 to 256 - as in the baseline. When doing so, the model simply transforms each frame into a speaker embedding expected to have information about the speaker active at that moment. These low-dimensional representations are then fed into the classification head, as shown in Figure 2b (left). The model is then trained with the same strategy as the original "per-segment" embedding extractor. Unlike the "per-segment" variant, the "per-frame" embedding extractor produces one embedding every 80,ms, determined by the ResNet stride, each labeled as some speaker in our case.




(a) Standard pipeline.

(b) Proposed pipeline.

Fig. 1: Comparison of modular diarization pipelines.




(a) Per-segment. (b) Per-frame (proposed).

Fig. 2: Standard and proposed embedding extraction. In the proposed approach, all embeddings are used to calculate the VAD loss, but only those corresponding to speech are used for the OSD loss. Speaker embeddings are denoted in cyan.

While the parameters of the model can be learned from a random initialization, one of the alternatives that we explored for training the per-frame embedding extractor was to initialize the parameters of the encoder with those of a model trained to produce per-segment embeddings (removing the pooling layer), and then retrain it to produce per-frame embeddings. The obtained model produced embeddings that allowed for similar diarization performance as training the model without a pooling layer from scratch; however, the convergence was much faster. Note that this is different from the teacher-student approach followed in [28], where a student model is trained to produce higher resolution embeddings (yet still using pooling) given an already trained teacher per-segment embedding extractor. In our case, we simply modify the original model and adapt it to produce higher-resolution embeddings.

### B. Integrated VAD and OSD

The second modification, seen in Figure 2b (right), is the addition of VAD and OSD "heads", which produce speech and overlap per-frame probabilities, respectively. Each per-frame embedding is passed through linear layers to produce the VAD and OSD logits. Both heads are trained using binary cross-entropy (BCE) loss. While all embeddings are used for the "VAD loss", only speech frames are used to calculate the "OSD loss", i.e., the head calculates the conditional probability that the frame is overlap given that it is speech.

In order to train the model in a multi-task fashion, the final loss to optimize is obtained as the weighted sum of the AAM, VAD BCE, and OSD BCE losses (with empirically obtained weights of 1, 5, and 2, respectively). Since VAD and OSD are a priori simpler tasks than speaker classification, there is no need to train the model on all tasks from the beginning. Besides, the same parameters are used for all tasks, but the model should use most of its potential for embedding extraction rather than VAD/OSD. Thus, we first train ResNet for per-frame embedding extraction. Only then the VAD and OSD heads are added and trained in a second training step for a few epochs. Training only VAD and OSD (i.e., without AAM loss) in the second stage can degrade speaker classification, so all three losses are necessary.

Another aspect is that the data used to train embedding extraction usually consist only of the speech of a single speaker (i.e., no silence or overlaps). Hence, using the same data for training the VAD and OSD heads is not possible. For this reason, we utilize a compound set of different corpora with diarization annotations to train the VAD and OSD losses. Since these datasets usually do not have absolute speaker labels and also contain only a limited number of speakers, they are not suitable for optimizing the embedding extraction loss. Conversely, speaker labels are not necessary to compute the VAD and OSD losses. Therefore, in the multi-task training, the VC2 data is used to calculate the AAM loss (Figure 2b left), and the compound set is used to calculate the VAD and OSD losses (Figure 2b right). Each loss is used to update all the parameters of the ResNet and its corresponding classification head. At inference time, there is a single forward pass that produces embeddings and VAD and OSD decisions. Note that this setup takes advantage of the supervision available for different tasks in a natural way and differs from EEND-like models which require all labels for all training data.

## IV. Experimental Setup

### A. Datasets

In order to evaluate the robustness of the proposed approach, we utilized two popular datasets: DIHARD II [38] and AMI [39] in its single distant microphone (SDM) version. Information about the datasets is presented in Table I. To train the model for VAD and OSD, a compound set was utilized, comprised of data from AISHELL-4 [40], AliMeeting [41] (mix far and near field), AMI [39], [42] (mix-headset and mix-array), DIHARD III [43], MSDWild [44], and RAMC [45]. For DIHARD III, where a train set was not available, the development set was included in the compound set; otherwise, the train set was used. The speaker embedding extractor was trained using the AAM loss on the VoxCeleb2 dataset [37], with 2290 hours of speech from 5994 speakers.

### B. Baseline and proposed method configurations

The baseline system uses VAD from pyannote [9] to identify speech regions. Per-segment embeddings are then extracted from these regions using 1.5 s segments with a 0.25 s stride. In the proposed system, per-frame embeddings are extracted every 0.08 s over the entire utterance, including silences. VAD—either from pyannote or a trained classification head—is then used to selects only speech-frame embeddings, which are then clustered by VBx following the recipe in [33].

For all systems, the embedding extractor and probabilistic linear discriminant analysis (PLDA) model needed by VBx were trained on VC2 data, where the training examples are 6s segments, randomly selected from the original VC2 utterances (original cuts). For PLDA training, one embedding was extracted from each such segment. In the per-frame system such embedding is selected from one random frame in the segment. DVBx [31] tuned on the development (or train, in the case of AMI) set was used to obtain VBx hyperparameters. As a final step, second speakers were assigned using pyannote's OSD and a heuristic that labels overlap regions based on neighboring segments [32].

## V. Results

Systems are evaluated in terms of diarization error rate (DER): the sum of missed speech (Miss), false alarm speech (FA), and confusion (Conf.) errors. VAD and OSD are evaluated in terms of misses and false alarms. All numbers are percentages. No forgiveness collar is used in any case.

### A. Per-frame embeddings

The first modification introduced is regarding the change from per-segment to per-frame embeddings. The comparisons are presented in Table II where system (1) is the per-segment baseline. System (2) is trained to produce per-frame embeddings starting from randomly initialized parameters, while system (3) starts from the parameters of (1), removing the pooling layer and randomly initializing the last linear layer before producing the embeddings. To evaluate all three approaches, pyannote VAD is utilized and it is possible to see that they perform very similarly. It should be pointed out that 9.7% out of the miss errors correspond to overlapped speech, which is not handled by these systems. While we expected the per-frame embeddings would lead to better results than the per-segment ones, the results did not necessarily prove this hypothesis. Per-frame embeddings represent only three times more frequent representations than the usual per-segment approach (80 ms vs. 250 ms) and it might be possible that such an increase is not enough to provide substantial gains. The performance may also be impacted by a mismatch between the PLDA backend, trained on embeddings from VC2 data containing almost no silence, and the test embeddings, which may capture non-speech regions due to the receptive field of the embedding extractor.

TABLE I: Statistics of evaluation and compound training sets.

| Dataset | Silence (%) | 1-speaker (%) | Overlap (%) | Hours |
|---|---|---|---|---|
| AMI (test) | 14.7 | 67.9 | 17.4 | 9.1 |
| DIHARD2 (eval) | 25.9 | 67.5 | 6.6 | 22.5 |
| Compound (train) | 24.3 | 55.0 | 20.7 | 774.2 |

Nevertheless, the proposed approach achieves a $4\times$ speedup compared to the standard per-segment approach by extracting all embeddings in a single pass. While the per-segment method processes short speech segments independently and requires multiple inference calls, the per-frame strategy performs embedding extraction together with joint VAD and OSD labeling.

### *B. Joint training*

System (4) in Table II investigates the use of simulated conversations (SC) [24], created from VC2 recordings, as training data with speaker, VAD, and OSD labels. The VAD performance was extremely poor. However, when pyannote VAD is used with the same embeddings, the DER drops from 39.5 to 26.5, with a confusion error of 6.9, indicating that the embedding quality is not compromised. Therefore, a fine-tuning step on real data was necessary for our VAD due to the significant mismatch between SC and real data. System (5) continues training the embedding extractor from (3), but now uses VC2 for the embedding extraction loss and a compound set of real datasets for the VAD and OSD losses. This model performs similarly to the models (1), (2), and (3) in the table. Small degradation in DER is mainly due to slightly worse VAD in comparison with pyannote's VAD possibly because each VAD is trained on a different compound set. The goal here is not to compare with pyannote, but to provide a point of reference. However, the results demonstrate that competitive performance can be achieved through joint training. Finally, system (6) is obtained by further fine-tuning system (5) using DIHARD dev set instead of the compound set to better match the VAD and overlap patterns of the test data.

Table III shows the results after applying overlap handling. System (7) corresponds to (1) with the heuristic applied using PyAnnote's OSD decisions, while (8) and (9) correspond to (5) and (6), respectively, using OSD outputs from the proposed model. In all cases, there is a mild improvement in the overall DER of the same order for these systems.

Finally, Table IV compares systems (1) and (5), as well as their respective versions with overlap handling of systems (7) and (8), on AMI (SDM), where the state-of-the-art performance with a more complex model is 18.9 [46]. We can see that the proposed approach reaches a similar performance as the baseline. VAD results are better than for DIHARD II, likely due to inclusion of the AMI training set in the compound data (for both baseline and proposed). This is supported by (6) and (9), obtained by fine-tuning (5) and (8) on DIHARD dev, yielding improved VAD.

## VI. Conclusions

In this work, we presented an approach to adapt a speaker embedding extractor for the purpose of diarization. Speaker embeddings were produced without a pooling mechanism while built-in mechanisms performed speech and overlap detection. With the proposed system, all three tasks are handled by a single model and the embedding extraction step is faster.

In spite of the competitive performance of the proposed approach, we believe that many options are yet to be explored:

- Use information from lower layers in the model since speech and overlap detection could better leverage it.
- Other architectures for speaker embedding extractors, with the developments on foundation models, training one model to perform different tasks at once should be possible.
- Finally, this work should not be understood as a means in itself but rather as an intermediate goal towards other applications. For example, for speaker verification, a system could automatically discard silence and overlap frames before producing (more robust) speaker embeddings without external VAD/OSD modules. Besides, we aim to combine the embedding extraction, speech, and overlap detection with DVBx [31] in order to have a full pipeline that can be trained (or fine-tuned) in an E2E manner.

TABLE II: Results on DIHARD II eval set. 'S' means per-segment embeddings, and 'F' means per-frame embeddings. 'Comp.' means compound set. 'FT' stands for finetuning.

| | System | | Diarization | | | | VAD | |
|---|---|---|---|---|---|---|---|---|
| Type | Train data | VAD | DER | Miss | FA | Conf. | Miss | FA |
| (1) S | VC2 | pyannote | 26.6 | 15.9 | 3.6 | 7.1 | 5.1 | 3.0 |
| (2) F | VC2 | pyannote | 26.8 | 15.9 | 3.6 | 7.3 | 5.1 | 3.0 |
| (3) S→F | VC2 | pyannote | 26.4 | 15.9 | 3.6 | 6.9 | 5.1 | 3.0 |
| (4) S→F | SC VC2 | joint | 39.5 | 12.7 | 17.4 | 9.4 | 2.4 | 14.3 |
| (5) S→F | VC2. + Comp. | joint | 26.9 | 16.2 | 4.0 | 6.7 | 5.3 | 3.3 |
| (6) S→F | FT VC2 + DH | joint | 26.1 | 15.0 | 3.5 | 7.5 | 4.4 | 2.9 |

TABLE III: Results after overlap post-processing on DIHARD II evaluation set. 'DH' stands for DIHARD II dev.

| | System | | Diarization | | | | OSD | |
|---|---|---|---|---|---|---|---|---|
| Type | Train data | VAD | DER | Miss | FA | Conf. | Miss | FA |
| (7) S | VC2 | pyannote | 26.2 | 14.2 | 4.5 | 7.5 | 5.2 | 0.7 |
| (8) S→F | VC2 + Comp. | joint | 26.6 | 15.3 | 4.5 | 6.8 | 5.9 | 0.4 |
| (9) S→F | FT VC2. + DH | joint | 25.8 | 14.4 | 3.8 | 7.6 | 6.1 | 0.2 |

TABLE IV: Comparison of per-segment embedding clustering (PyAnnote VAD/OSD) and per-frame embedding clustering (joint VAD/OSD) on AMI dataset.

| | Diarization | | | | VAD | | OSD | |
|---|---|---|---|---|---|---|---|---|
| System | DER | Miss | FA | Conf. | Miss | FA | Miss | FA |
| (1) | 35.6 | 25.4 | 2.0 | 8.2 | 10.2 | 1.9 | 11.7 | 0.0 |
| (7) | 33.9 | 21.5 | 2.9 | 9.5 | | | 8.1 | 0.8 |
| (5) | 36.1 | 21.7 | 2.2 | 12.2 | 6.7 | 2.1 | 11.7 | 0.0 |
| (8) | 34.8 | 17.1 | 4.1 | 13.6 | | | 7.4 | 1.8 |


## Acknowledgment

The work was partly supported by Ministry of Education, Youth and Sports of the Czech Republic (MoE) through the OP JAK project "Linguistics, Artificial Intelligence and Language and Speech Technologies: from Research to Applications" (ID:CZ.02.01.01/00/23_020/0008518), and by Czech Ministry

of Interior project No. VJ01010108 "ROZKAZ". Computing on IT4I supercomputer was supported by MoE through the e-INFRA CZ (ID:90254).

## REFERENCES


[1] G. Sell *et al.*, "Diarization is Hard: Some Experiences and Lessons Learned for the JHU Team in the Inaugural DIHARD Challenge." in *Interspeech*, 2018, pp. 2808–2812.

[2] F. Landini *et al.*, "BUT System for the Second DIHARD Speech Diarization Challenge," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6529–6533.

[3] T. J. Park *et al.*, "Auto-tuning spectral clustering for speaker diarization using normalized maximum eigengap," *IEEE Signal Processing Letters*, vol. 27, pp. 381–385, 2019.

[4] Y. Fujita *et al.*, "End-to-End Neural Speaker Diarization with Permutation-Free Objectives," in *Proc. Interspeech*, 2019.

[5] S. Horiguchi *et al.*, "End-to-End Speaker Diarization for an Unknown Number of Speakers with Encoder-Decoder Based Attractors," in *Proc. Interspeech*, 2020, pp. 269–273.

[6] I. Medennikov *et al.*, "Target-Speaker Voice Activity Detection: A Novel Approach for Multi-Speaker Diarization in a Dinner Party Scenario," in *Proc. Interspeech*, 2020, pp. 274–278.

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

[8] ——, "Advances in Integration of End-to-End Neural and Clustering-Based Diarization for Real Conversational Speech," in *Proc. Interspeech*, 2021, pp. 3565–3569.

[9] H. Bredin, "pyannote.audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe," in *Proc. INTERSPEECH 2023*, 2023.

[10] A. Plaquet and H. Bredin, "Powerset multi-class cross entropy loss for neural speaker diarization," in *Proc. INTERSPEECH 2023*, 2023.

[11] F. Landini, "From Modular to End-to-End Speaker Diarization," Ph.D. Thesis, Brno University of Technology, Faculty of Information Technology, 2024. [Online]. Available: https://www.fit.vut.cz/study/phd-thesis/1357/

[12] S. Baroudi *et al.*, "pyannote. audio speaker diarization pipeline at VoxSRC 2023," *The VoxCeleb Speaker Recognition Challenge 2023 (VoxSRC-23)*, 2023.

[13] F. Landini *et al.*, "DiaPer: End-to-End Neural Diarization with Perceiver-Based Attractors," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[14] D. Snyder *et al.*, "X-vectors: Robust DNN embeddings for speaker recognition," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[15] K. He *et al.*, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[16] H. Zeinali *et al.*, "But system description to voxceleb speaker tecognition challenge 2019," *arXiv preprint arXiv:1910.12592*, 2019.

[17] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," in *Proc. Interspeech 2020*, 2020.

[18] J. Thienpondt and K. Demuynck, "ECAPA2: A hybrid neural network architecture and training strategy for robust speaker embeddings," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023.

[19] T. J. Park, M. Kumar, and S. Narayanan, "Multi-scale speaker diarization with neural affinity score fusion," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7173–7177.

[20] Y. Kwon *et al.*, "Multi-scale speaker embedding-based graph attention networks for speaker diarisation," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8367–8371.

[21] Y. J. Kim *et al.*, "Advancing the dimensionality reduction of speaker embeddings for speaker diarisation: disentangling noise and informing speech activity," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

[22] J.-w. Jung *et al.*, "In search of strong embedding extractors for speaker diarisation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

[23] J.-H. Choi *et al.*, "Efficient Speaker Embedding Extraction Using a Twofold Sliding Window Algorithm for Speaker Diarization," in *Interspeech 2024*, 2024, pp. 3749–3753.

[24] F. Landini *et al.*, "From Simulated Mixtures to Simulated Conversations as Training Data for End-to-End Neural Diarization," in *Proc. Interspeech 2022*, 2022, pp. 5095–5099.

[25] N. Yamashita, S. Horiguchi, and T. Homma, "Improving the Naturalness of Simulated Conversations for End-to-End Neural Diarization," in *Proc. The Speaker and Language Recognition Workshop (Odyssey 2022)*, 2022, pp. 133–140.

[26] F. Landini *et al.*, "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.

[27] V. A. Miasato Filho, D. A. Silva, and L. G. D. Cuozzo, "Multi-objective Long-Short Term Memory Neural Networks for Speaker Diarization in Telephone Interactions," in *2017 Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE, 2017, pp. 181–185.

[28] T. Cord-Landwehr *et al.*, "Frame-wise and overlap-robust speaker embeddings for meeting diarization," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[29] Y. Kwon *et al.*, "Look who's not talking," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 567–573.

[30] J. Thienpondt and K. Demuynck, "Speaker Embeddings With Weakly Supervised Voice Activity Detection For Efficient Speaker Diarization," in *Proc. The Speaker and Language Recognition Workshop (Odyssey 2024)*, 2024, pp. 131–136.

[31] D. Klement *et al.*, "Discriminative Training of VBx Diarization," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 11 871–11 875.

[32] S. Otterson and M. Ostendorf, "Efficient use of overlap information in speaker diarization," in *2007 IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*. IEEE, 2007, pp. 683–686.

[33] F. Landini *et al.*, "Bayesian HMM Clustering of x-vector Sequences (VBx) in Speaker Diarization: Theory, Implementation and Analysis on Standard Tasks," *Computer Speech & Language*, vol. 71, 2022.

[34] S. Wang *et al.*, "Advancing speaker embedding learning: Wespeaker toolkit for research and production," *Speech Communication*, vol. 162, p. 103104, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167639324000761

[35] J. Deng *et al.*, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[36] X. Xiang *et al.*, "Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2019, pp. 1652–1656.

[37] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep Speaker Recognition," in *Proc. Interspeech 2018*, 2018, pp. 1086–1090.

[38] N. Ryant *et al.*, "Second DIHARD challenge evaluation plan," *Linguistic Data Consortium, Tech. Rep*, 2019.

[39] J. Carletta *et al.*, "The AMI meeting corpus: A pre-announcement," in *International workshop on machine learning for multimodal interaction*. Springer, 2006, pp. 28–39.

[40] Y. Fu *et al.*, "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.

[41] F. Yu *et al.*, "M2MeT: The ICASSP 2022 multi-channel multi-party meeting transcription challenge," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6167–6171.

[42] W. Kraaij *et al.*, "The AMI meeting corpus," in *Proc. International Conference on Methods and Techniques in Behavioral Research*, 2005.

[43] N. Ryant *et al.*, "The Third DIHARD Diarization Challenge," in *Proc. Interspeech 2021*, 2021, pp. 3570–3574.

[44] T. Liu *et al.*, "MSDWild: Multi-modal Speaker Diarization Dataset in the Wild," in *Proc. Interspeech 2022*, 2022, pp. 1476–1480.

[45] Z. Yang *et al.*, "Open Source MagicData-RAMC: A Rich Annotated Mandarin Conversational(RAMC) Speech Dataset," in *Proc. Interspeech 2022*, 2022, pp. 1736–1740.

[46] M. Härkönen, S. J. Broughton, and L. Samarakoon, "Eend-m2f: Masked-attention mask transformers for speaker diarization," in *Interspeech 2024*, 2024, pp. 37–41.