TAL TECH

DOCTORAL THESIS

Improved Training Methods for Multi-Talker Speech Processing

Joonas Kalda

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY
TALLINN 2026

TALLINN UNIVERSITY OF TECHNOLOGY
DOCTORAL THESIS
15/2026

Improved Training Methods for Multi-Talker Speech Processing

JOONAS KALDA

TAL TECH PRESS

Tallinn University of Technology
School of Information Technologies
Department of Software Science

The dissertation was accepted for the defence of the degree of Doctor of Philosophy (Applied Something) on 18 February 2026

Supervisor: Professor Tanel Alumäe,
Department of Software Science,
School of Information Technologies,
Tallinn University of Technology,
Tallinn, Estonia

Opponents: Professor Anthony Larcher,
Le Mans University,
Le Mans, France

Oldřich Plchot, PhD,
Brno University of Technology,
Brno, Czech Republic

Defence of the thesis: 13 March 2026, Tallinn

Declaration:
Hereby I declare that this doctoral thesis, my original investigation and achievement, submitted for the doctoral degree at Tallinn University of Technology, has not been submitted for any academic degree elsewhere.

Joonas Kalda

signature

ISSN 2585-6898 (publication)
ISBN 978-9916-80-465-0 (publication)
ISSN 2585-6901 (PDF)
ISBN 978-9916-80-464-3 (PDF)
DOI https://doi.org/10.23658/taltech.15/2026

Kalda, J. (2026). *Improved Training Methods for Multi-Talker Speech Processing* [TalTech Press]. https://doi.org/10.23658/taltech.15/2026

TALLINNA TEHNIKAÜLIKOOL
DOKTORITÖÖ
15/2026

Treeningmeetodid mitme rääkijaga kõne töötluseks

JOONAS KALDA

TAL TECH
KIRJASTUS

Contents

List of publications

The present Ph.D. thesis is based on the following publications that are referred to in the text by Roman numbers.

I Joonas Kalda and Tanel Alumäe. Collar-aware training for streaming speaker change detection in broadcast speech. In *Odyssey 2022: The Speaker and Language Recognition Workshop*, pages 141–147, 2022

II Joonas Kalda, Clément Pagés, Ricard Marxer, Tanel Alumäe, and Hervé Bredin. PixIT: Joint training of speaker diarization and speech separation from real-world multi-speaker recordings. In *Odyssey 2024: The Speaker and Language Recognition Workshop*, pages 115–122, 2024

III Joonas Kalda, Tanel Alumäe, Martin Lebourdais, Hervé Bredin, Séverin Baroudi, and Ricard Marxer. TalTech-IRIT-LIS speaker and language diarization systems for DISPLACE 2024. In *25th Annual Conference of the International Speech Communication Association, Interspeech 2024*, pages 1635–1639, 2024

IV Joonas Kalda, Séverin Baroudi, Martin Lebourdais, Clément Pagés, Ricard Marxer, Tanel Alumäe, and Hervé Bredin. Design choices for PixIT-based speaker-attributed ASR: Team ToTaTo at the NOTSOFAR-1 challenge. *Computer Speech & Language*, page 101824, 2026

V Joonas Kalda, Clément Pagés, Tanel Alumäe, and Hervé Bredin. Diarization-Guided multi-speaker embeddings. In *26th Annual Conference of the International Speech Communication Association, Interspeech 2025*, pages 5233–5237, 2025

Author's contributions to the publications

I I was the main author of this work. I contributed to defining the methodology, implementing the model, running the experiments, analyzing the results, and co-writing the paper.

II I was the main author of this work. I contributed to defining the methodology, implementing the model, running the experiments, analyzing the results, and

writing the majority of the paper.

III I was the main author of this work. I contributed to defining the methodology, preparing the data, running the experiments, analyzing the results, and writing the majority of the paper.

IV I was the main author of this work. I contributed by defining the methodology, implementing the model, running the experiments, analyzing the results, and writing the majority of the paper.

V I was the main author of this work. I contributed by defining the methodology, implementing the model, running the experiments, analyzing the results, and writing the majority of the paper.

Chapter 1

Introduction

The field of automatic speech processing has achieved remarkable success, particularly in controlled, single-speaker environments [6, 7]. However, a significant gap remains in processing the speech of multiple individuals in natural settings such as meetings, interviews, and broadcast media [8]. This complexity is best exemplified by the "cocktail party problem" [9], where the acoustic scene contains multiple concurrent speakers, environmental noise, and temporally overlapping speech. Effectively processing such recordings involves determining speaker identity ("who spoke?"), temporal localization ("when did they speak?"), and isolating speech content ("what did they say?").

This chapter introduces the domain of multi-talker speech processing (MTSP), a research area encompassing interconnected computational tasks for the analysis, segmentation, and attribution of speech in complex multi-talker environments. These tasks aim to extract speaker-specific information across a spectrum of detail, from fine-grained temporal analysis (e.g., detecting precise speaker turn changes) to comprehensive scene understanding (e.g., complete diarization and speech separation). Robust MTSP systems are critical for applications such as multi-talker automatic speech recognition (ASR), meeting summarization, speech-to-speech translation, voice anonymization, and speech enhancement. In ASR and meeting summarization, a failure to correctly attribute utterances to speakers can result in fundamentally flawed records of the conversation. Similarly, in systems for speech-to-speech translation and voice anonymization, inaccurate speaker attribution can corrupt identity mapping, leading to mismatched voices, inconsistent context, or the unintended disclosure of a speaker's identity.

In this chapter, the core computational tasks within MTSP are outlined. This is followed by a discussion of the challenges that motivate the work in this thesis: domain mismatch between training and deployment conditions, the difficulty of processing overlapping speech, and the ambiguity in human annotations. These challenges serve as the motivation for the training methods proposed in this thesis.

1.1 Underlying tasks

1.1.1 Speaker diarization

Speaker diarization (SD) addresses the question "who spoke when?" by partitioning an audio stream into temporal segments that are homogeneous by speaker identity and assigning a unique label to each segment [10, 11]. Modern systems typically operate via either a multi-stage pipeline [12] or an end-to-end neural network [13].

Multi-stage pipelines traditionally comprise the following sequential subtasks:

1. **Voice activity detection (VAD):** Distinguishes regions containing speech from non-speech audio.

2. **Speaker change detection (SCD):** Pinpoints the precise temporal moments when the active speaker changes.

3. **Speaker embedding extraction:** Converts each uniform-speaker speech segment into a fixed-size numerical vector (an embedding) that captures the unique acoustic characteristics of the speaker's voice.

4. **Speaker clustering:** Groups the extracted embeddings based on similarity to determine the number of unique speakers and assign a consistent identity label to each segment. Common techniques include spectral or Bayesian clustering methods [14].

A major limitation of this traditional pipeline is its inability to effectively handle overlapping speech. This led to the emergence of end-to-end neural diarization (EEND) models [13, 15], which learn to generate the speaker activity timeline directly from the audio in a single step. By framing diarization as a multi-label prediction problem, EEND systems inherently accommodate overlapping speech, as they can simultaneously predict activity for multiple speakers.

However, EEND systems face their own challenges. They often suffer from high computational and memory costs, making them difficult to apply to long-duration recordings. Initial models were also restricted to a predefined, fixed number of speakers, although later extensions like EEND-EDA [15] addressed this limitation. Furthermore, because a single training sample encompasses an entire recording, these models rely heavily on synthetically generated conversations.

To combine the strengths of both paradigms, hybrid approaches such as EEND-vector clustering (EEND-VC) have been developed [16, 17]. These systems utilize an EEND model to perform overlap-aware diarization on short, localized audio segments, replacing the separate VAD and SCD steps of the traditional pipeline. EEND-VC benefits from the precise, overlap-aware segmentation of EEND while retaining the scalability and speaker-agnostic nature of clustering-based methods for long recordings. This thesis further explores the use of such clustering-based strategies,

particularly concerning their integration with speaker embedding extraction and speech separation.

1.1.2 Speaker change detection

Speaker change detection (SCD) is the task of identifying the exact temporal points in an audio stream where the speaker identity changes. Conceptually, this is a sequence labeling task, where the system predicts the probability of a speaker change event for each time frame. Beyond its role as a component in traditional SD pipelines, SCD is also applied independently in scenarios such as automatic subtitling where explicit speaker clustering is not required [18].

The underlying assumption for SCD is that the audio can be cleanly segmented into regions of single-speaker activity (i.e., speakers do not overlap). This assumption is crucial for the subsequent embedding extraction step in multi-stage SD [12]. The quality of SCD is crucial, as imprecise segment boundaries can result in mixed-speaker segments that degrade the accuracy of speaker embeddings. Modern SCD approaches often leverage deep neural networks that learn to detect changes directly from the audio's spectral features [19].

1.1.3 Speaker recognition

Speaker recognition (SR) focuses on analyzing the acoustic characteristics of a speech segment to determine a speaker's identity [20, 21]. The central component of modern SR is the creation of **speaker embeddings**: a function is learned to map a variable-length speech segment to a fixed-size numerical vector that represents the speaker's unique voice characteristics. This function is trained to maximize inter-speaker discriminability (ensuring embeddings from different speakers are distant) and minimize intra-speaker variability (ensuring embeddings from the same speaker are closely clustered).

These embeddings underpin three primary applications:

- **Speaker verification:** A one-to-one comparison task to confirm or deny the claim that two speech segments originate from the same speaker.
- **Speaker identification:** A one-to-many task that assigns a speech segment to the most probable speaker from a known set of enrolled individuals.
- **Speaker clustering:** An unsupervised task that groups a collection of speech segments into clusters, where each cluster corresponds to a distinct speaker. This is also the final step of the multi-stage diarization approach.

The evolution of SR has moved from generative models, such as gaussian mixture models with universal background models (GMM-UBM) [22] and factor analysis approaches like i-vectors [23], to discriminative deep neural architectures. Contemporary state-of-the-art systems employ encoders like x-vectors [24], ECAPA-TDNN

[25], and ResNet variants [26], and are typically trained on massive speaker verification datasets using angular margin losses [27]. Despite achieving stellar performance in clean, single-speaker conditions, these models exhibit significant performance degradation when deployed in challenging multi-speaker environments [28].

1.1.4 Speech separation

Speech separation is arguably the most comprehensive task in MTSP, aiming to solve the "cocktail party problem" by decomposing a mixed audio signal containing multiple concurrent speakers into individual, clean audio streams for each speaker.

Contemporary deep learning methodologies, including Conv-TasNet [29], dual-path RNN (DPRNN) [30], and transformer-based models [31, 32], have achieved remarkable separation performance. The standard approach to training and evaluating these systems relies solely on synthetic data created by artificially mixing clean, single-speaker recordings. This is because obtaining ground-truth clean sources for real-world recordings is practically impossible, as individual headset microphones inevitably pick up cross-talk during a conversation. However, this creates a significant domain mismatch when models are applied to real-world conversations, which are characterized by complex room acoustics, microphone effects, and realistic conversational dynamics such as interruptions and backchannels [33, 34]. To mitigate this and enable generalization to real-world, unsupervised methods like mixture invariant training (MixIT) [35] have been introduced. MixIT bypasses the need for clean source signals by training the model on "mixtures of mixtures" and calculating the loss with respect to the original mixtures. This approach presents its own set of difficulties, such as a proneness to over-separation and achieving reliable inference on long-form audio.

1.2 Fundamental challenges and research gaps

It is clear that despite substantial progress in individual MTSP tasks, several fundamental challenges persist. The research presented in this thesis concentrates on a few of these, focusing mainly on the themes of annotation ambiguity and domain mismatch.

1.2.1 Annotation ambiguity and evaluation mismatch

Manually annotating the precise start and end times of speech segments is an inherently challenging and subjective task. Human annotators routinely disagree on the exact location of a speaker change due to factors such as inter-turn silences, subtle speech overlaps, and the limits of human perception. This disagreement typically falls within a range of 100 to 500 ms. Further discrepancies can arise from variations in dataset-specific annotation guidelines.

To account for this subjectivity, standard evaluation protocols for tasks like SD and SCD employ a "forgiveness collar". A predicted change point is deemed correct if it falls within a tolerance window (e.g., ±250 ms) of a ground-truth annotation [36].

However, a critical mismatch arises during model training. Most models are trained using standard point-wise loss functions (e.g., binary cross-entropy) that penalize any predictions that do not perfectly match the single ground-truth frame. This training objective entirely ignores the temporal tolerance permitted in evaluation, potentially yielding models that are overly brittle and sub-optimally tuned for the final evaluation metric.

1.2.2 Domain mismatch

A second fundamental bottleneck across many MTSP tasks is the significant disparity between the data used for training and the real-world conditions where models are deployed.

Reliance on synthetic data for separation. As discussed, speech separation models are predominantly trained on artificially mixed audio. This synthetic environment fundamentally lacks the genuine acoustic properties (e.g., reverb, noise) and realistic conversational dynamics present in real recordings [33]. Consequently, models trained exclusively on synthetic data often fail to generalize effectively to authentic multi-speaker conversations.

Single-speaker training for speaker embeddings. Speaker embedding models, which are central to robust diarization, face a similar domain mismatch. They are typically trained on vast collections of single-speaker utterances (e.g., VoxCeleb [37]). Yet, in real-world applications, these models must process speech segments that are frequently contaminated by concurrent speech from other speakers, which severely degrades embedding quality and reduces speaker discriminability [28]. Furthermore, current systems require that each speaker in an overlapping segment be processed sequentially. Thus, a significant speed-up and a potential improvement in accuracy could be achieved by learning to process these concurrent speakers simultaneously.

1.3 Thesis outline and contributions

This thesis addresses the aforementioned challenges by exploring a set of targeted methodologies designed to enhance the robustness and practical utility of MTSP systems. The primary contributions are:

- **Aligning training and evaluation objectives** by developing a novel collar-aware loss function for speaker change detection that directly incorporates the

temporal tolerance for annotation errors used in standard evaluation protocols (Chapter 2).

- **Bridging domain gaps in separation and diarization** by designing a joint training methodology that leverages real-world, multi-speaker recordings to simultaneously learn speaker diarization and long-form speech separation, building upon and extending the MixIT line of work (Chapter 3).
- **Enhancing robustness and speed of overlap processing** by proposing a technique to guide speaker embedding models with diarization information, adapting them for multi-speaker audio and enabling simultaneous processing of concurrent speakers (Chapter 4).

Chapter 2

Collar-aware training for speaker change detection

This chapter is based on the work in publication I.

As discussed in Section 1.2.1, a critical mismatch exists between the training and evaluation of speaker change detection (SCD) systems. While evaluation protocols commonly use a "forgiveness collar" to account for the inherent ambiguity in manual annotations, models are typically trained with loss functions that penalize any deviation from a single ground-truth frame. This discrepancy can result in models that are not optimally tuned for the metric by which they are judged. This chapter introduces a novel training method designed to directly address this issue by incorporating the concept of a tolerance collar into the training objective itself.

2.1 Background

SCD is a task of locating precise points in an audio recording when a different speaker starts speaking. It is often used as the first step in speaker diarization systems. Depending on the application, SCD systems can be either streaming (also known as *online*) or batch-processing (*offline*). In a batch processing system, the whole audio recording is available when the SCD system is applied. This allows the model to use all information from both past and future frames when locating speaker change points. A streaming model, on the other hand, needs to identify speaker change points with low latency, using typically only one or two seconds of audio from the future. Streaming SCD is needed as a preprocessing step in streaming speech recognition systems that perform unsupervised speaker adaptation, e.g. using i-vectors [38], so that the speaker adaptation state could be reset at speaker change points. SCD is also often an explicit requirement in realtime closed captioning systems for broadcast television [18].

Most modern SCD systems are based on supervised learning. Large speech datasets, manually annotated with speaker change points, are used for training and SCD is treated as a binary sequence classification task. Long short-term memory (LSTM) recurrent neural networks [19, 39] or convolutional neural networks [40–42] are often used as models. An important issue when training such models for SCD is that the annotated change points in the training data are ambiguous and imbalanced. The ambiguousness comes from the fact that often there is a substantial amount of silence between the speech of two adjacent speakers, yet only a single frame is marked as a change point. The choice where exactly the annotated change point resides is often inconsistent, resulting in training data that is confusing for the model. Also, the number of frames in the training data labelled as change points is usually less than 1% of all the frames, causing problems with model convergence.

This chapter proposes a novel objective function for training sequence classification models for SCD. This *collar-aware* objective function gives the SCD model more freedom by allowing it to choose an appropriate speaker change point within the neighbourhood of the annotated change point. This method addresses both the problems of imbalanced data as well as the ambiguousness of the annotated labels. Furthermore, the models trained using this method are especially well suited for streaming applications, as the resulting model generates "peaky" change points that do not require any post-processing to find local maxima. We show that the method also achieves notably higher accuracy in both streaming and batch-processing scenarios, compared to several well-established baselines[1].

2.2 Related work

SCD approaches can be divided into two main categories: metric- and model based. The first approach operates by applying a pair of sliding windows on the sequence of feature vectors extracted from the underlying audio signal and uses a divergence metric for comparing their contents. A speaker change point is detected if the divergence between two adjacent windows is larger than a predefined threshold and the divergence achieves a significant local maximum. The advantage of this method is that it doesn't require a large annotated training corpus for training: only the value of the threshold parameter needs finetuning on a small validation set. This method is used in many speaker diarization systems that use Gaussian mixture models (GMMs) as their main building blocks (e.g. [43])

A model based approach, on the other hand, uses a a training corpus with manually annotated speaker change points to train a model for this task. Many different models have been proposed, such as hidden Markov models [44], GMMs [45], eigenvoices [46], deep neural networks (DNNs) [42, 47], convolutional neural

[1]Code and demo available at `https://github.com/alumae/online_speaker_change_detector`

networks [40–42], recurrent neural networks [19, 39] and Siamese networks [48].

Since models based on neural networks have become popular in recent years for this task, we review three approaches based on them more carefully. In [19], SCD is formulated as a standard binary sequence labelling task that can be tackled using bidirectional LSTMs: the model's task is to label each frame with either 0 (no speaker change) or 1 (speaker change). One problem with this approach is that the training data is heavily imbalanced: the number of frames that are labelled with 0 is much larger than the number of frames labelled with 1 (only 0.4% according to [19]). Under standard training, the model converges to a state in which a 0 is predicted for each frame. To address this, the approach in [19] increases the number of positive labels artificially by labelling frames 50 ms on each side of the annotated change point as 1. During inference, local score maxima exceeding a pre-determined threshold are marked as speaker change points.

In [42], a somewhat similar approach is used, but instead of a bidirectional LSTM, a CNN is used that "sees" a fixed-size window of feature frames prior and after the current frame. This allows operating the model with low latency in streaming mode. As with the LSTM-based approach, a large number of frames in the direct neighbourhood of the annotated change point are labelled as positive during model training, in order to make the training data more balanced.

In [48], a Siamese architecture is used for low-latency SCD: a 2-second window prior and after the current frame is processed by a bidirectional LSTM, resulting in two embedding vectors. The embeddings are then processed by a classification module that decides whether the two segments correspond to different speakers. Various pretraining schemes can be applied to the embedding computation module that are found to improve the detection performance by a large amount. This work handles the imbalanced data problem by sampling a predefined ratio of speaker change points from the training data to each batch.

Inconsistent and unreliable speaker turn boundaries in manually annotated training data can also have a negative effect on the performance of end-to-end speaker diarization systems. In [49], a modification to the standard multilabel classification loss for speaker diarization is introduced that simply ignores the errors in a defined radius around annotated speaker change points.

2.3 Collar-aware training

Speaker change detection is often regarded as a binary sequence labelling problem. We consider an audio recording consisting of feature vectors x_i for $i = 1, ..., N$ and the corresponding speaker boundary labels $y_i \in \{0, 1\}$ with $y_i = 1$ meaning that the frame corresponds to an annotated speaker boundary.

When a SCD system is evaluated in terms of precision and recall of detected speaker boundaries, it is a standard practice to use a *collar* (typically 250 ms) for

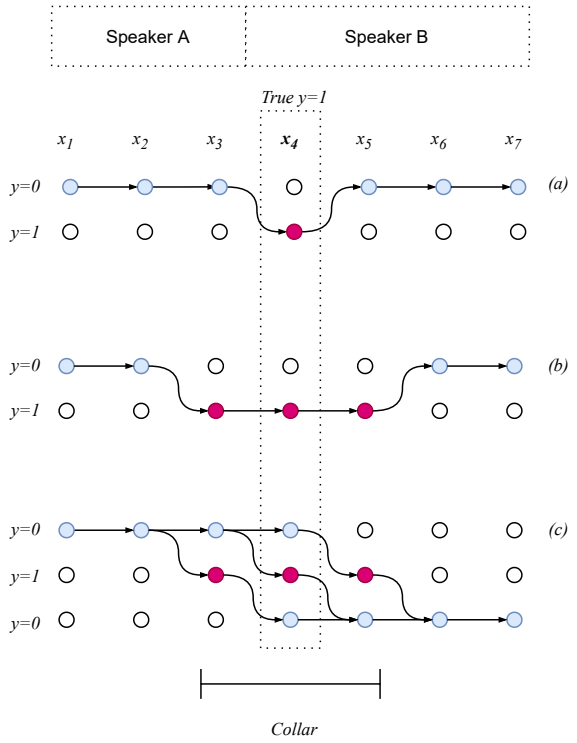

Figure 2.1: ***Outline of three supervision methods for speaker change detection:*** *(a) corresponds to standard sequence labelling objective; (b) increases the number of positive labels artificially by setting several frames in the neighbourhood of the annotated change point as positive [19]; (c) the proposed method sums over all paths that have exactly one positive label in the neighbourhood of the annotated change point.*

annotated speaker boundaries: if the boundary detected by the model is within the tolerated amount of milliseconds of the annotated boundary, the detected speaker change point is assumed to be correct. However, under standard sequence labelling objective (Figure 2.1, *a*), the collar is not used, making the training objective different from the evaluation scenario.

As pointed out in the previous section, some approaches [19, 42] have suggested to artificially modify the training data of the speaker boundary detection model by labelling a predefined number of frames around the annotated speaker boundary as additional (pseudo-)boundaries (Figure 2.1, *b*). This is done in order to make the training data more balanced in terms of label frequencies, and to model the inherent ambiguousness of the speaker boundaries.

We propose to use a modified objective function for training SCD models that solves both the problems of imbalanced data and ambiguous annotated boundaries. Instead of labelling points around the annotated boundary as pseudo-boundaries, it supervises the model to label exactly one frame within the given collar as a speaker boundary, but the exact position of the boundary can be freely chosen (Figure 2.1,

```python
def collar_bce_loss(log_probs, change_points, collar):
  """
  Compute collar-aware binary CE loss.

  Arguments:
  log_probs -- tensor of shape (seq_len, 2), containing log likelihoods of non-boundary
    and boundary events
  change_points -- indexes of annotated boundaries
  collar -- value of the collar (in frames)
  """
  result = log_probs[:, 0].sum()
  for change_point in change_points:
    collar_variant_logs = []
    collar_start_i = change_point - collar
    collar_end_i = change_point + collar
    time_index = range(collar_start_i, collar_end_i + 1)
    event_index = torch.eye(collar_end_i - collar_start_i + 1).long()
    collar_variant_logprobs = log_probs[time_index, event_index].sum(1)
    result -= log_probs[time_index, 0].sum()
    result += torch.logsumexp(collar_variant_logprobs, 0)
  return -result
```

Figure 2.2: PyTorch code for efficient calculation of the collar-aware binary cross-entropy loss.

c). This method has several advantages: (1) it matches the evaluation criteria better than method (*b*); (2) it solves the imbalanced data problem similarly or better than method (*b*); (3) the model trained in this manner can be easily applied in online mode without any post-processing to find the local maximum, since the output of the model is now very "peaky" (see Section 2.5.1).

Formally, given the reference labels y and model predictions $\hat{y}_i$, the standard binary sequence labelling objective is:

$$\mathcal{L}(\hat{y}, y) = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)$$

Since the boundaries occur very sparsely, this objective can be efficiently calculated by summing over the log likelihoods of the no-boundary events, and then modifying it to account for the few boundary events. Given annotated boundary positions $Z = \{z | y_z = 1\}$, the standard sequence labelling loss becomes:

$$\mathcal{L}(\hat{y}, Z) = -\Big(\sum_{i=1}^{N} \log(1 - \hat{y}_i) - \sum_{z_i \in Z} \log(1 - \hat{y}_{z_i}) + \sum_{z_i \in Z} \log(\hat{y}_{z_i}) \Big)$$

In order to calculate the proposed collar-aware objective, we have to consider a superset $S(Z)$ of all sets of boundary events Z' where for each original change point $z_i \in Z$ there is exactly one boundary event that is within it's collar set $C_i = \{x | z_i - c < x < z_i + c\}$, where c is the value of the collar. Alternatively,

$$S(Z) = \{\{z_1, ..., z_N\} | z_i \in C_i \forall i \in \{1, ..., N\}\},$$

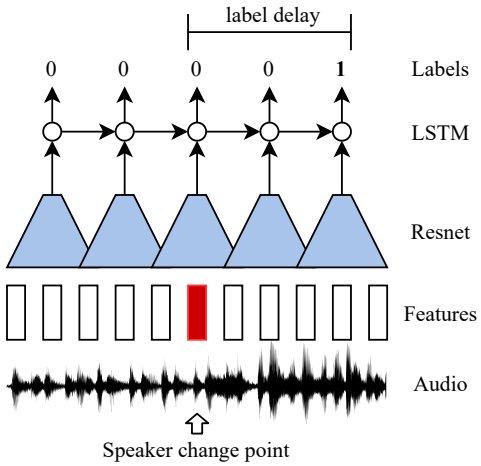

Figure 2.3: ***Architecture of the streaming Resnet-LSTM model:*** *Filterbank features from speech frames are fed into a Resnet module that is pretrained on a speaker recognition task and then trained jointly with the rest of the model. The outputs from the Resnet module are fed into a LSTM layer that identifies speaker boundaries (*`label` $= 1$*). Since this cannot be done without the knowledge of future frames, the identification is done with a label delay that corresponds to 1 second of speech (in streaming mode).*

where N is the total number of original change points. For example, if the reference label sequence is [00100], then Z' at collar $= 2$ corresponds to a set of label sequences $\{[01000], [00100], [00010]\}$.

The proposed objective sums over all such change point configurations:

$$\mathcal{L}_{\text{collar}}(\hat{y}, Z) = -\log \sum_{Z' \in S(Z)} \mathrm{e}^{-\mathcal{L}(\hat{y}, Z')}$$

The idea of this objective function is somewhat similar to the CTC loss function [50] used for training end-to-end speech recognition models. As with CTC, it is not practical to compute it using brute force. In order to make it more efficient, we can again use our knowledge that the number of speaker boundaries occur very sparsely. Figure 2.2 lists the PyTorch implementation of this idea. The collar-aware loss can be calculated by summing over the log-likelihoods of the non-boundary events (line 11), subtracting the log likelihoods of the non-boundary events that lie within a boundary collar (line 19), and then adding the marginalized log-likelihood of having exactly one boundary somewhere within the collar (line 20). While this approach is computationally more expensive than the standard binary cross-entropy loss, this overhead is limited to the training phase and does not impact the model's inference speed.

Table 2.1: A comparison of lengths and the number of speaker change points in the datasets used in the experiments.

Dataset	Train	Development	Test
Estonian	497.2h / 80k	1.2h / 166	0.7h / 102
English	128.3h / 19.5k	6.1h / 893	5.4h / 893

2.4 Experiments

2.4.1 Datasets

The experiments were carried on both English and Estonian datasets. For English, we used the HUB4 speech dataset [51, 52]. The Estonian dataset consists of TV and radio broadcasts. Both datasets are manually transcribed and annotated with speaker information. Test and development data were separated similarly for both datasets: 10 recordings were chosen for each at random. An overview of dataset sizes and annotated boundary counts is provided in Table 2.1. The datasets are similarly balanced, with 0.04% of the frames being labelled as speaker change points.

2.4.2 Implementation details

We consider two different architectures which we train using both the standard training method and the proposed collar-aware one.

The first architecture was chosen to closely resemble that of [19]. 33-dimensional acoustic features are extracted every 10ms on a 25ms window, consisting of 11-dimensional MFCCs and their first and second derivatives. The model is made up of two Bi-LSTM layers having 64 outputs and 40 outputs and a multi-layer-perceptron with 40-, 10- and 1-dimensional layers.

The second architecture uses a Resnet-based feature extractor before a LSTM layer. The Resnet module is extracted from a speaker recognition model pretrained on VoxCeleb2 [53], as described in [54]. It results in 1280-dimensional features with a frame subsampling rate of 8. In the low-latency streaming model, the Resnet layer is followed by two 256-dimensional LSTM layers, and 1-second label delay is used in order for the model to see the data past the current frame (see Figure 2.3). In the offline model, the LSTMs are replaced with bidirectional LSTMs, and no label delay is used.

All our training methods use extracted segments with random lengths between 10s and 30s.

The first training method used also follows [19]. Namely the training data is artificially modified by positively labelling every frame in a 50ms neighborhood of an annotated change point. Notably, no additional labelling is needed for the Resnet-based architecture since the subsampling that happens during feature extraction results in frames of 80ms duration and thus a 50ms neighborhood corresponds to

Table 2.2: Precision (P), recall (R) and F1 results of various batch-mode and streaming models on Estonian and English datasets with two two different forgiveness collar values.

	Estonian dataset						English dataset					
	collar=0.25s			collar=0.50s			collar=0.25s			collar=0.50s		
Model	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Batch-mode processing												
Pretrained speaker diarization (VBx)	0.68	0.68	0.68	0.96	0.96	**0.96**	0.48	0.64	0.55	0.67	0.88	0.76
Pretrained *pyannote.audio*	0.62	0.73	0.67	0.68	0.79	0.73	0.42	0.38	0.40	0.57	0.51	0.54
+ finetuned on the given dataset	0.82	0.82	0.82	0.89	0.89	0.89	0.60	0.49	0.54	0.73	0.59	0.65
BLSTM	0.7	0.85	0.77	0.74	0.88	0.81	0.44	0.59	0.50	0.50	0.62	0.55
+ collar aware training	0.75	0.81	0.78	0.86	0.80	0.83	0.59	0.57	0.58	0.61	0.61	0.61
Resnet + BLSTM	0.80	0.78	0.79	0.84	0.80	0.82	0.59	0.66	0.62	0.65	0.69	0.67
+ collar aware training	0.92	0.89	**0.91**	0.96	0.92	0.94	0.76	0.69	**0.73**	0.79	0.76	**0.78**
Streaming processing												
pyannote.audio with latency=1.0s	0.34	0.67	0.45	0.37	0.73	0.49	0.21	0.33	0.26	0.28	0.44	0.34
+ finetuned on our data	0.42	0.68	0.51	0.46	0.75	0.57	0.26	0.45	0.32	0.30	0.52	0.38
Resnet + LSTM	0.73	0.73	0.73	0.76	0.75	0.76	0.56	0.62	0.59	0.58	0.71	0.64
+ collar aware training	0.89	0.83	**0.86**	0.92	0.86	**0.89**	0.66	0.71	**0.68**	0.72	0.75	**0.74**

roughly a single frame. A standard binary sequence labelling objective is used as the loss function for this method.

The second training method includes no artificial labelling and instead uses the proposed collar-aware objective as the loss function. The size of the collar was chosen to be $c = 250$ms and the effects of varying the collar size are discussed in Section 2.5.2.

During training, data augmentation is applied: background noise and/or reverberation is added to each training segment, both with a probability of 0.3. The background noises originate from the MUSAN corpus [55]. For reverberation, we used simulated small and medium room impulse responses [56] and real room impulse responses from the BUT Speech@FIT Reverb Database [57].

2.4.3 Evaluation metrics

The evaluation metrics are standard precision (P), recall (R), and F1-score, calculated on the test sets. Predicted change points are considered correct if they match an annotated change point within a forgiveness collar (closest pairs are matched first until no pairs remain). Although our main evaluation metrics are precision and recall of detected speaker change points at a forgiveness collar of 250 ms, we also show the same metrics using a larger 0.5 second collar. A change point is predicted to happen if the local maximum of the models output is higher than a threshold, the value of which is determined by maximizing the F1 score on the development set.

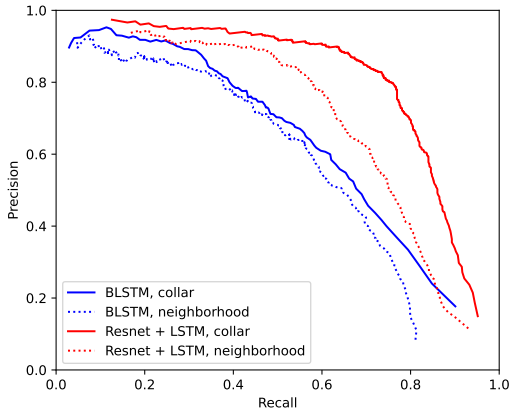

Figure 2.4: Precision-recall curves for models trained on the English dataset.

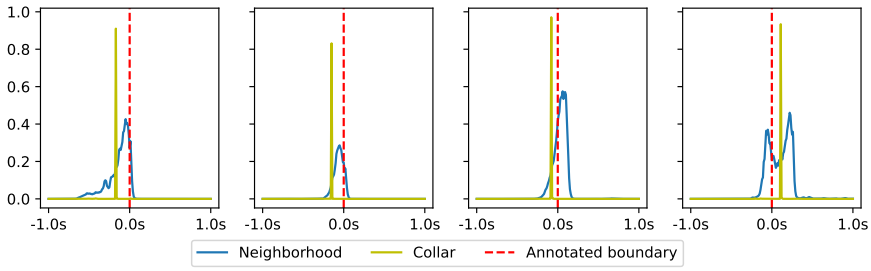

Figure 2.5: Random samples of Resnet+BLSTM model outputs for neighborhood- and collar-based models trained on the English dataset centered around annotated speaker change points.

2.4.4 Baselines

In addition to the pure LSTM-based speaker segmentation system [19], we compare our results to various baselines. Since speaker change points can be easily derived from the output of a speaker diarization system, we use several speaker diarization models that have achieved competitive results on various diarization benchmarks.

The recently proposed VBx diarization method [14] has produced state-of-the-art results on CALLHOME, AMI and DIHARD II datasets. The method uses a Bayesian hidden Markov model to find speaker clusters in a sequence of x-vectors. We used the open source implementation of the method available at GitHub[2]. The diarization pipeline first extracts x-vectors from the sections of the audio that contain speech. The provided x-vector models are trained on VoxCeleb1 [37],

[2] `https://github.com/BUTSpeechFIT/VBx`

VoxCeleb2 [53] and CN-CELEB [58]. The x-vectors are extracted every 0.25 seconds from overlapping sub-segments of 1.5 seconds. The x-vectors are centered, whitened and length normalized [59]. The x-vectors are pre-clustered using agglomerative hierarchical clustering to obtain the initial speaker labels and finally further clustered using the VBx model. The used the VBx parameters F_A, F_b and P_{loop} tuned on the respective development sets in order to minimize the boundary detection F1 score with a 250 ms forgiveness collar.

We also compare to the neural speaker segmentation method implemented in *pyannote.audio* [60] that performs joint voice activity detection, speaker segmentation and overlapped speech detection. Similarly to the original EEND approach [13], here speaker segmentation is modeled as a multi-label classification problem using permutation-invariant training. The model operates on short audio chunks (5 seconds) at a temporal resolution of every 16 ms and outputs speaker activation probabilities that are stitched together across frames. More specifically, we use the model available at `https://huggingface.co/pyannote/segmentation` that is trained on the DIHARD3 corpus [61]. We also experiment with the same model in a low-latency setting [62], using the open-source implementation[3]. In streaming mode, the latency of the segmentation output is configurable. To make the results comparable to our streaming model, we used a latency of 1 second.

In addition to using the publicly available *pyannote.audio* segmentation model, we also experimented with finetuning it on our training data. This was done on each dataset and resulted in further baselines for both streaming and offline settings.

In order to convert the output of the diarization systems to speaker change points we consider all the consecutive pairs of speaker segments where the speaker ids differ. If there is less than 2 seconds between the two segments then a speaker change point is predicted at the beginning of the second segment. This was found to give better results than other choices in the gap like the midpoint or the end of the first segment.

2.5 Results

A comparison of the model performances on the two datasets is provided in Table 2.2. The results are divided into two categories: models that perform change point detection in batch mode, and streaming models. The Resnet+LSTM based models trained using the proposed collar-aware loss function clearly outperform the same models trained using the standard training method on both datasets. Furthermore, these models also provide higher speaker change point detection accuracy than the baseline speaker diarization models. The state-of-the-art VBx diarization model actually results in impressive accuracy at a collar of 0.5 seconds but much lower accuracy at the standard 0.25 second collar. This might be due to the fact that the

[3] `https://github.com/juanmc2005/StreamingSpeakerDiarization/`

VBx model uses a relatively large temporal resolution of 0.25 seconds which causes the detected change point to be considered an error if it is off by just one timestep.

Precision-recall curves obtained by varying the classification threshold on the English test set are presented in Figure 2.4. It can be seen that collar-aware training outperforms neighbourhood-based training at all operation points for both LSTM and Resnet-BLSTM based models.

2.5.1 Peakiness of model output

One benefit of the collar-aware loss-function discussed above was the "peaky" output of the model. Figure 2.5 demonstrates this effect by visualizing samples obtained from Resnet+BLSTM model outputs centered around randomly chosen annotated boundaries for the English dataset. The output obtained from a model trained using the neighborhood-based method is spread out over multiple frames requiring finding the exact local maximum in post-processing. In comparison, the change points predicted by the model trained with the collar-based method can be obtained by simply comparing the model outputs to a threshold since the activations tend to be limited to a single frame.

2.5.2 Tuning the collar size

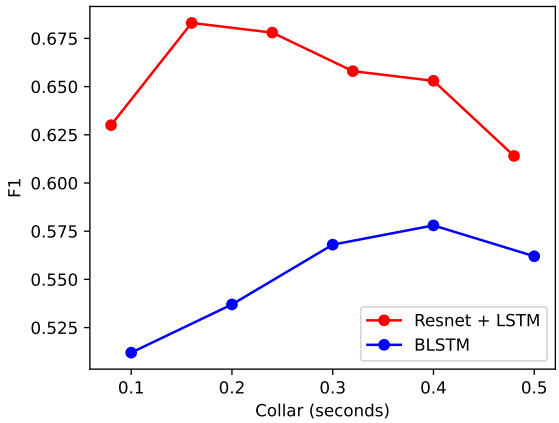

Figure 2.6: F1 scores on the English dataset for models of varying training collar size.

Figure 2.6 shows the influence of the collar size used during training on the F1 score on English test data. The optimal size for the collar is dependant on the nature of the data, how imbalanced it is and how reliable the annotated boundaries are. Overall, there seems to be flexibility to the choice of collar size as the F1 score does not change a lot across the tested range. Notably, all of the tested collar sizes lead to a better result than the neighborhood based models.

2.6 Conclusion

This chapter presented a novel supervision method for speaker change detection models using a collar-aware objective function. Our experiments compared it with a conventional training method, which artificially labels a neighborhood of an annotated boundary as positive, as well as with various state-of-the-art speaker diarization models. We found that our collar-aware training yields improved results for both a purely LSTM-based model and one that uses pretrained embeddings with 8-fold subsampling.

The exact choice of collar size was found to not have a great effect on performance, with choices from 80 ms to 500 ms all outperforming the conventional training method.

We analyzed model outputs around randomly chosen boundaries and showed that the activations for our method are concentrated to a single frame. This makes our training method well-suited for online applications, as it eliminates the need for local maxima detection in post-processing.

Due to these benefits, the introduced SCD system was deployed as part of an online closed captioning system for the Estonian Parliament and the Estonian Public Television [63].

Chapter 3

Joint training of speaker diarization and speech separation

This chapter is based on the work presented in publications II, III, and IV.

Processing real-world, multi-speaker audio for applications like speaker-attributed automatic speech recognition (SA-ASR) presents significant challenges, primarily due to overlapping speech. While speaker diarization identifies active speakers and speech separation isolates their voices, conventional training paradigms limit their real-world applicability. Supervised separation systems suffer from a domain mismatch because they rely on synthetic training data, and unsupervised alternatives like mixture invariant training (MixIT) often over-separate a single speaker's voice into multiple output streams. This chapter introduces PixIT, a joint training framework that combines these two tasks. The main contribution is a multi-task objective that uses permutation invariant training (PIT) for diarization and MixIT for separation, requiring only speaker diarization labels to train on real-world recordings. This approach not only improves separation quality but also enhances diarization by enabling the extraction of speaker embeddings from the cleaned, separated audio. This chapter details architectural explorations, the critical impact of matched ASR fine-tuning, and PixIT's successful performance on several competitive benchmarks.

3.1 Background

Speech separation—the task of isolating individual speakers' voices from mixed audio signals—is crucial for downstream tasks such as speaker-attributed automatic speech recognition (SA-ASR). The predominant training paradigm for deep learning-

based speech separation has been supervised learning, typically using permutation invariant training (PIT) with synthetically mixed clean audio sources [29, 30]. However, models trained on synthetic data often fail to generalize to real-world recordings due to the inherent domain mismatch, as clean, isolated sources are rarely available in practice.

To overcome the reliance on synthetic data, unsupervised methods have been developed. Mixture invariant training (MixIT) is a prominent unsupervised approach that trains on real-world recordings by creating a "mixture-of-mixtures" (MoM) from two original mixtures and training a model to separate them [35]. A key limitation of MixIT is that the separation model must handle twice the number of speakers present in a single mixture, which often leads to over-separation—a single speaker's voice being split across multiple output channels—when applied to standard mixtures during inference.

Another significant challenge is processing long-form audio. Separation models are typically trained on short segments, and applying them to long recordings requires a stitching mechanism. Continuous speech separation (CSS) is a common technique that applies a separation model on a sliding window and stitches the outputs based on source similarity in the overlap regions [64]. However, this approach can fail if a speaker is silent for a period longer than the window overlap, necessitating a separate speaker diarization system to maintain long-term speaker identity.

The complementary nature of speech separation and speaker diarization has inspired several joint training approaches. Models like the recurrent selective attention network (RSAN) [65] and end-to-end neural diarization and speech separation (EEND-SS) [66] have shown promise. However, these methods still depend on synthetic data for training their separation components, which prevents them from fully bridging the gap to real-world data.

This chapter introduces PixIT, a joint training framework that addresses these limitations by combining PIT for speaker diarization with MixIT for speech separation. PixIT leverages real-world recordings, requiring only speaker diarization labels for training. By constraining the number of speakers in the MoMs, this method mitigates the over-separation problem inherent in MixIT. A core benefit of PixIT is that it produces separated audio sources that are temporally aligned with speaker activity predictions, which simplifies inference on long-form audio and allows for effective post-processing.

The effectiveness of PixIT is validated in several contexts. It substantially improves SA-ASR performance on challenging meeting datasets, such as the NOTSOFAR-1 Challenge [67]. Furthermore, its strong diarization capabilities are demonstrated in the DISPLACE 2024 Challenge, where using PixIT-separated sources for speaker embedding extraction contributed to the winning system's performance.

The main contributions detailed in this chapter are:

- The proposal of PixIT, a novel framework for jointly training speaker diarization and speech separation on real-world data using a combined PIT and MixIT loss.
- An in-depth analysis of architectural choices for the joint model, including alternative self-supervised learning (SSL) features and advanced masking networks.
- A demonstration that fine-tuning an ASR system on PixIT-separated sources significantly boosts downstream SA-ASR performance, surpassing fine-tuning on original mixtures.
- A novel approach for improving speaker diarization by extracting speaker embeddings from separated sources, validated in the DISPLACE 2024 Challenge.
- A comprehensive evaluation showing PixIT's competitiveness against strong baselines, including CSS-based systems, on multiple public benchmarks.
- The release of open-source recipes to facilitate further research and reproducibility[1].

3.2 Methodology

Our model is based on the TasNet architecture [29], which consists of a 1-D convolutional encoder, a separator module that predicts N masking matrices, and a 1-D convolutional decoder. We also leverage pre-trained WavLM features [68], which are particularly well-suited for speech separation due to their pre-training with an utterance mixing data augmentation strategy. These features are concatenated with the convolutional encoder outputs. The diarization network takes the encoded separated signals as input and processes each source independently to perform what is effectively voice activity detection (VAD). This independent processing of sources is required to maintain alignment between the separation outputs and the diarization branches. The joint model architecture, which we call ToTaToNet[2], is illustrated in Figure 3.1. Components of the model related to the **diarization** branch are colored **orange**, components related to **separation** are colored **purple**, and components used by **both branches** are colored with a **gradient** between the two. This color scheme is kept consistent across all figures in this chapter.

[1] `https://github.com/joonaskalda/PixIT` and `https://github.com/joonaskalda/PixIT-design-choices`

[2] A name reflecting the collaboration between labs in **To**ulouse, **Ta**llinn, and **To**ulon.

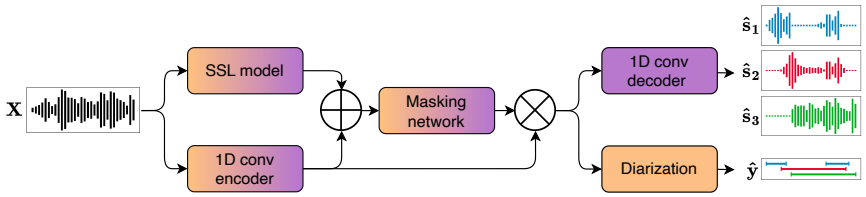

Figure 3.1: The architecture of the proposed ToTaToNet model.

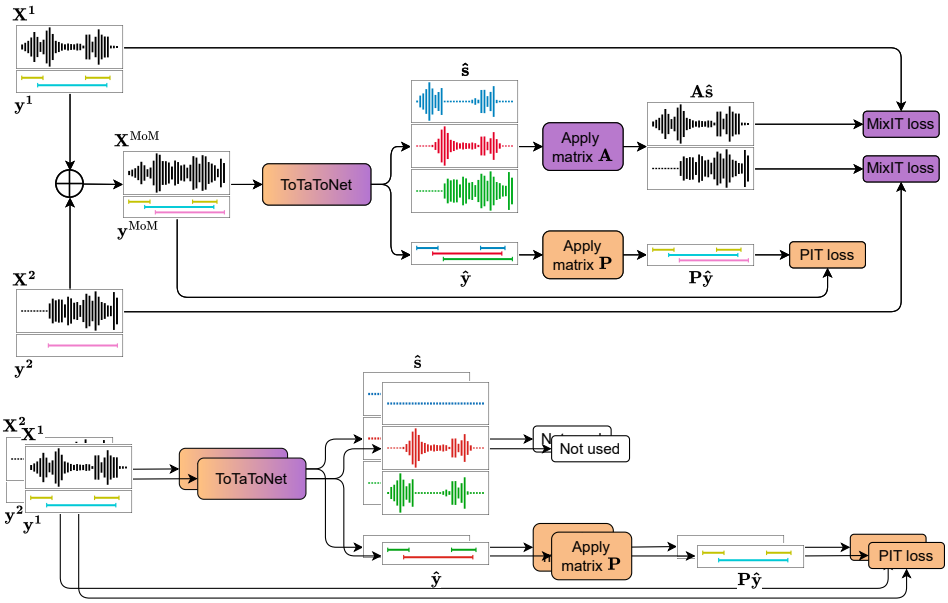

Figure 3.2: Training the joint model. The upper part shows the calculation of the MixIT and PIT losses on MoMs. The bottom part shows the calculation of PIT losses on the original mixtures.

3.2.1 Training

The joint training method for speech separation and speaker diarization is illustrated in Figure 3.2. Consider an audio chunk $\mathbf{X}$ and the reference speaker activity labels $\mathbf{y} \in \{0,1\}^{K_{\max} \times T}$, where $y_{k,t} = 1$ if speaker k is active at frame t, and $y_{k,t} = 0$ otherwise. Here, $K_{\max}$ specifies the maximum number of speakers anticipated in an audio chunk. For diarization, we use the well-established PIT objective [13]:

$$\mathcal{L}_{\text{PIT}}(\mathbf{y}, \hat{\mathbf{y}}) = \min_{\mathbf{P}} \sum_{k=1}^{K_{\max}} \mathcal{L}_{\text{BCE}}\left(\mathbf{y_k}, [\mathbf{P}\hat{\mathbf{y}}]_k\right),$$

where $\hat{\mathbf{y}}$ are the predicted speaker activations, $\mathbf{P}$ is a $K_{\max} \times K_{\max}$ permutation matrix, and $\mathcal{L}_{\text{BCE}}$ is the standard binary cross-entropy loss.

Using the speaker annotations, we construct two audio chunks, $(\mathbf{X^1}, \mathbf{y^1})$ and $(\mathbf{X^2}, \mathbf{y^2})$, containing non-overlapping sets of speakers, with the total number of

speakers being no greater than $K_{\max}$. Limiting the total number of speakers is critical for solving the over-separation issue of MixIT. The MoM is constructed as $\mathbf{X}^{\text{MoM}} = \mathbf{X^1} + \mathbf{X^2}$, and the corresponding speaker activity labels $\mathbf{y}^{\text{MoM}}$ are formed by concatenating the labels of active speakers from both chunks so that $\mathbf{y}^{\text{MoM}} \in \{0,1\}^{K_{\max} \times T}$. The MixIT loss function is then given by:

$$\mathcal{L}_{\text{MixIT}}\left(\{\mathbf{X_n}\}, \hat{\mathbf{s}}\right) = \min_{\mathbf{A}} \sum_{n=1}^{2} \mathcal{L}_{\text{SI-SDR}}\left(\mathbf{X_n}, [\mathbf{A}\hat{\mathbf{s}}]_n\right),$$

where $\hat{\mathbf{s}}$ are the predicted separated sources, M is the number of output sources, $\mathbf{A}$ is a mixing matrix $\mathbf{A} \in \{0,1\}^{2\times M}$ under the constraint that each column sums to 1, and $\mathcal{L}_{\text{SI-SDR}}$ is the negative scale-invariant signal-to-distortion ratio [69].

Our combined multi-task loss is:

$$\mathcal{L}_{\text{PixIT}} = \lambda\Big(\mathcal{L}_{\text{PIT}}(\mathbf{y^1}, \hat{\mathbf{y}}^{\mathbf{1}}) + \mathcal{L}_{\text{PIT}}(\mathbf{y^2}, \hat{\mathbf{y}}^{\mathbf{2}}) + \mathcal{L}_{\text{PIT}}(\mathbf{y}^{\text{MoM}}, \hat{\mathbf{y}}^{\text{MoM}})\Big) + (1-\lambda)\mathcal{L}_{\text{MixIT}}\left(\{\mathbf{X_n}\}, \hat{\mathbf{s}}\right),$$

where $\lambda = 0.5$ was selected from 0.1, 0.5, 0.9 based on its superior performance on the development data. While a thorough sensitivity analysis of the λ hyperparameter was not conducted, the optimal balance between diarization and separation losses likely relates to dataset characteristics, such as the degree of speech overlap.

3.2.2 Inference

During inference, an audio stream is partitioned into shorter chunks, as depicted in Figure 3.3. The joint model processes each chunk and outputs aligned estimates for speaker sources and speaker activations. The resulting speaker activations and sources are clustered as in [70]. First, speaker activations are binarized using a detection threshold $\theta \in [0,1]$ to identify speaker segments. Second, a local speaker embedding is extracted for each active speaker in a chunk. For this, we use only the regions of the chunk where the corresponding speaker is active. Speaker embeddings are computed by feeding the concatenation of original audio samples from these regions to the pre-trained ECAPA-TDNN model [25] available in [71]. Finally, agglomerative hierarchical clustering is performed on these embeddings using a clustering threshold δ.

As an important post-processing step, we perform leakage removal by setting the stitched separated sources at time t to zero when the diarization output indicates that the corresponding speaker is not active within a window $[t - \Delta t, t + \Delta t]$. This is a key benefit of having aligned speaker activations and sources, as it eliminates cross-talk when a speaker is inactive. The goal of introducing Δt is to provide downstream ASR systems with additional context. The hyperparameters θ, δ, and Δt are optimized on the development dataset.

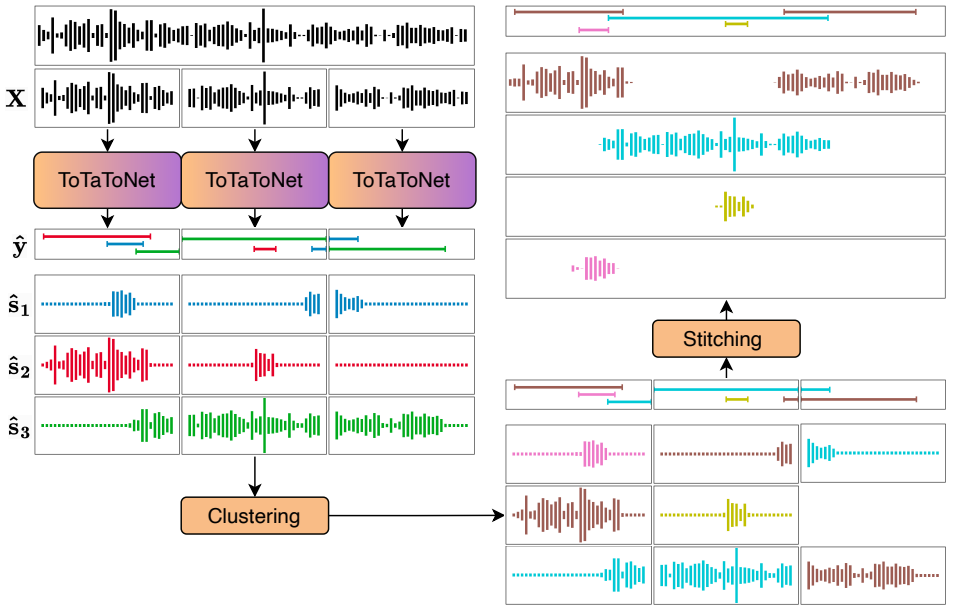

Figure 3.3: Inference on long-form audio. For ease of visualization, inference using non-overlapping sliding windows is shown.

3.2.3 SSL features

The ToTaToNet architecture can incorporate features from pre-trained self-supervised learning (SSL) models to improve its representations. We evaluate two such models. The first is WavLM (large version) [68], which is particularly suited for this work as its pre-training includes an utterance mixing data augmentation strategy that helps the model learn to handle overlapping speech.

We compare this to a more recent and larger model, the Conformer-based W2v-BERT 2.0 [72] from the Seamless project [73]. It features a larger architecture (580M vs. 315M parameters for WavLM-large), combines a masked language modeling (MLM) objective with a contrastive loss, and was pre-trained on a significantly larger and more diverse multilingual dataset (4.5 million hours vs. 96k hours). For the remainder of this study, we use WavLM to refer to its large version.

3.2.4 Masking networks

The masking network is a core component of the ToTaToNet separation module. The original implementation uses the dual-path recurrent neural network (DPRNN) [30]. DPRNN processes encoded audio features by segmenting them into chunks and applying recurrent layers both within chunks (intra-chunk) and across chunks (inter-chunk) to model local and global dependencies.

To potentially improve separation performance, we also evaluate a more recent architecture: the monaural speech separation transformer 2 (MossFormer2) [74].

MossFormer2 replaces the sequential processing of RNNs with self-attention mechanisms to better capture global context. It is composed of stacked blocks containing local and global self-attention layers and a feed-forward sequential memory network (FSMN) [75] to model long-range dependencies more effectively than traditional RNNs. This change comes at a computational cost, with MossFormer2 having a significantly larger parameter count (20-50M vs. 2-3M for DPRNN) and longer training times.

3.2.5 ASR fine-tuning

Although modern ASR models achieve impressive performance on general speech recognition tasks, their effectiveness often deteriorates in domain-specific scenarios that differ from their training data. Therefore, ASR models are often fine-tuned on in-domain data to improve accuracy. This adaptation is particularly crucial for multi-party meeting scenarios due to significant divergences from standard ASR training distributions, such as speaker overlap, variable signal-to-noise ratios, reverberation, and non-stationary background noises. Multi-party conversations also feature non-uniform speaker turns, frequent interruptions, and context dependencies spanning multiple turns.

Fine-tuning adapts both acoustic and language models to these domain-specific phenomena. A primary challenge in multi-party meeting transcription is handling overlapping speech. Contemporary ASR architectures typically cannot generate parallel token streams for simultaneous speakers and instead process overlapping segments sequentially, ordered by utterance onset. This sequential processing can complicate the accurate determination of word timestamps.

This challenge is specific to single-channel audio containing multiple overlapping speakers. While speech separation preprocessing can mitigate this issue, it introduces its own challenges. When processing separated streams, the ASR system must contend with reduced contextual information, as each stream contains only the speech of a single participant. This limitation can impact the model's ability to leverage broader conversational context. Additionally, isolating individual speakers complicates the processing of short backchannel utterances, which often derive meaning from their temporal relationship to other speakers' contributions.

While both single-stream and separated-stream approaches have distinct challenges, it is essential to maintain consistency between the preprocessing methods applied during training and inference. Specifically, if speaker separation is used during inference, the training data must be processed similarly. During the training phase, separation can be guided using reference information, such as speaker-attributed word timestamps. In some cases, individual microphone recordings are available, providing "gold standard" separated streams. In our experiments, we investigate this consistency hypothesis by evaluating ASR models fine-tuned on both single-stream and separated-stream configurations across both testing conditions.

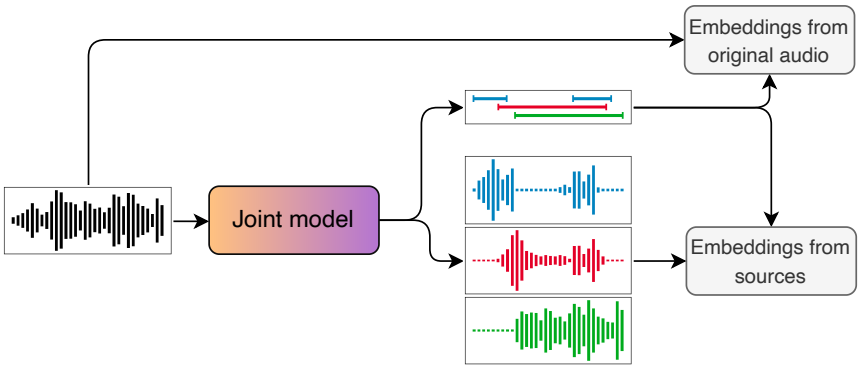

Figure 3.4: Speaker embedding extraction using either the active frames from separated sources or the original audio as input.

3.2.6 Separated sources as input to speaker embeddings

During inference, clustering local speaker segments requires extracting a representative speaker embedding for each active speaker in a given chunk. As illustrated in Figure 3.4, these embeddings can be extracted either from the original mixture audio or from the corresponding separated source signal generated by PixIT. Each approach presents a trade-off. Extracting embeddings from the original audio is challenging in regions with overlapping speech, as this can corrupt the speaker representation. Using the separated source mitigates this by providing an overlap-free signal. However, the separation process may introduce audio artifacts that could degrade the quality of the speaker embedding.

This work systematically evaluates this trade-off by comparing diarization and SA-ASR performance when using both embedding extraction methods across multiple datasets. The clustering hyperparameters are tuned independently for each approach to ensure a fair comparison.

3.2.7 Tackling ASR timestamp errors caused by long silent regions

PixIT's file-level separated sources often contain substantial periods of silence. This issue is particularly pronounced in the NOTSOFAR-1 dataset, where meetings can have up to eight speakers. During our challenge participation, we used faster-whisper[3], a reimplemented Whisper decoder that incorporates VAD to remove silent regions before processing the audio. However, this approach introduced a timing issue, as Whisper-assigned word timestamps could fall on the incorrect side of a VAD boundary. When this happens, the final timestamps can be shifted by the duration of the removed silent region, which leads to large timing errors. These misalignments can increase the tcpWER by falling outside the acceptable time

[3] `https://github.com/SYSTRAN/faster-whisper`

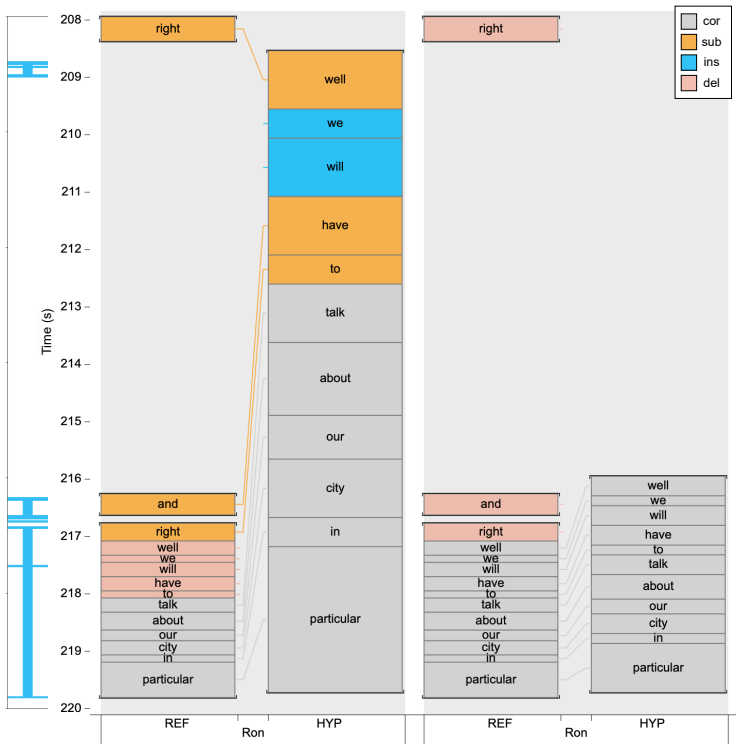

Figure 3.5: An example of timestamp errors caused by long silences. From left to right: Active speech segments predicted by PixIT, speaker-attributed ASR output's tcpWER alignment visualized before timestamp refinement, and after refinement.

collar, as illustrated in Figure 3.5.

Faster-whisper includes a heuristic to mitigate this issue, but utterances can still be split between distant VAD segments. Instead of using Silero VAD[4], as in faster-whisper, we use active speech segments from PixIT's diarization output. This approach incurs no additional computational cost and better aligns the VAD train-test domains. To account for artifacts introduced by PixIT's separation, we further refine the heuristic to adjust timestamps only when diarization detects inactivity for more than half of the utterance's duration.

3.3 Experiments

3.3.1 Datasets

Our experiments use three distinct and publicly available datasets, all sourced from single-microphone meeting recordings. The first is AMI [76], which consists of 100 hours of recordings from multi- and single-channel microphones across 171

[4] `https://github.com/snakers4/silero-vad`

meetings. The dataset includes both scenario-driven and natural meetings. The second is AliMeeting [77], a Mandarin corpus with approximately 120 hours of natural meeting recordings across 212 sessions.

While AMI contains approximately 15–20% overlapping speech, AliMeeting presents more challenging scenarios with around 40% overlap. Finally, the NOTSOFAR-1 [67] dataset contains 150 hours of single-channel recordings and 110 hours of multi-channel recordings, totaling 280 meetings. The dataset also includes 1000 hours of tailored synthetic mixtures. Due to computational limitations, we do not use this synthetic data for training PixIT. While AMI and AliMeeting contain recordings of approximately 30 to 60 minutes, NOTSOFAR-1 is composed of shorter, 6 to 7-minute files. Since PixIT is a single-channel method, our experiments focus on AMI-SDM (single distant microphone), AliMeeting *channel 1*, and NOTSOFAR-SC.

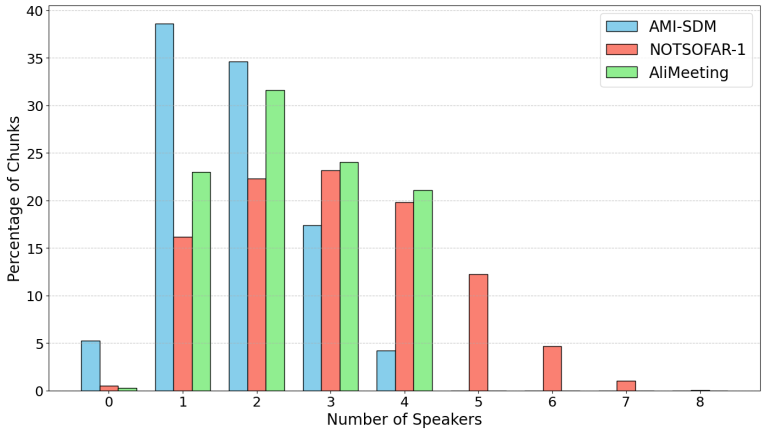

Figure 3.6: Histogram of the total number of speakers per 5-second chunk across the training sets of the datasets.

As shown in Figure 3.6, AMI-SDM contains the fewest speakers per 5-second chunk, making it the least challenging of the three datasets. In contrast, NOTSOFAR-1 is the most challenging for both diarization and separation due to a high number of active speakers per chunk. Furthermore, the large number of recorded meetings adds significant diversity to the scenarios.

3.3.2 Evaluation metrics

Our systems were evaluated on speaker diarization and speaker-attributed transcription. For speaker diarization, we employed the diarization error rate (DER), a standard metric computed as:

$$DER = \frac{FA + MISS + SC}{TOTAL}, \tag{3.1}$$

where FA (false alarm) is the duration of non-speech wrongly classified as speech, MISS (missed detection) is the duration of speech incorrectly classified as non-speech, and SC (speaker confusion) is the duration of speech assigned to the wrong speaker. TOTAL is the sum of reference speech durations for all speakers, counting overlapped speech multiple times. We report DER with a 0-second collar.

To evaluate speaker-attributed transcription, we rely on the concatenated minimum-permutation word error rate (cpWER) [78] and the time-constrained minimum-permutation WER (tcpWER) from MeetEval [79]. Both are extensions of the word error rate (WER), which is defined as:

$$WER = \frac{I + S + D}{S + D + C}, \tag{3.2}$$

where I, S, D, and C are the number of inserted, substituted, deleted, and correct words, respectively.

The cpWER penalizes speaker confusion errors. To compute it, reference and hypothesis segments are grouped by speaker and then concatenated. The Hungarian algorithm [80] is then used to find the permutation of speakers that minimizes the WER. Transcripts from unmatched speakers (either in the hypothesis or the reference) are counted as errors.

The tcpWER adds a temporal constraint to penalize matching words that are far apart, thereby evaluating the quality of the temporal prediction. In our experiments, we use a temporal collar of 5 seconds, consistent with the NOTSOFAR-1 Challenge.

For text normalization, we used Whisper's normalizer on AliMeeting and AMI, and the slightly modified version from the NOTSOFAR-1 challenge on its respective dataset. Metrics were aggregated by summing the individual components across all files, except for NOTSOFAR-1, where we averaged the metric values per file to match the challenge evaluation protocol.

3.3.3 Speaker attribution methods

The standard approach for adding speaker attribution to an off-the-shelf ASR system is to integrate it with a speaker diarization system, as illustrated in Figure 3.7. Each ASR speech segment is assigned to the speaker who is most active during that segment according to the diarization output. A CSS system can be used as a preprocessing step to better handle overlapping speech, but this requires tailored synthetic training data.

The NOTSOFAR-1 challenge baseline includes such a CSS-based system, which we use for comparison. Since no publicly available synthetic datasets exist that are tailored for AMI or AliMeeting, we also use the NOTSOFAR-1 baseline system for comparison on those datasets.

PixIT offers a more integrated approach. It outputs long-form separated sources that are inherently speaker-attributed through their alignment with the diarization

Table 3.1: Overview of parameter counts, training hyperparameters, and real-time factors (RTFs) for the proposed ToTaToNet architectures. RTF is measured on the AMI development set.

SSL	Masking network	# Params		Batch size	Learning rate		RTF
		Frozen	Trainable		SSL	Other	
WavLM	DPRNN	0	319M	16	1e-5	3e-4	0.005
WavLM	MossFormer2	0	319M	8	1e-5	3e-4	0.009
W2v-BERT	DPRNN	580M	5M	16	1e-5 (LoRA)	3e-4	0.012
W2v-BERT	MossFormer2	580M	21M	8	1e-5 (LoRA)	3e-4	0.017

output. Consequently, performing SA-ASR with PixIT simply involves passing each long-form source directly to an ASR system. We use these two SA-ASR methods to benchmark PixIT's separation capabilities, as a direct evaluation is not possible on real-world audio due to the lack of clean reference signals.

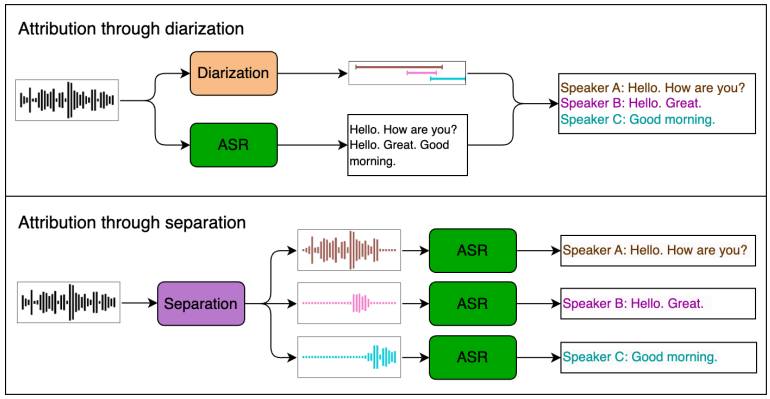

Figure 3.7: Speaker attribution for ASR via either diarization or separation.

3.3.4 Implementation details

All models were optimized using the Adam optimizer [81] and employ 3 output masks. The WavLM and W2v-BERT components were fine-tuned with a learning rate of 1e-5, while the remainder of the model used a learning rate of 3e-4. Given the size of W2v-BERT, we adopted LoRA (low-rank adaptation) with a rank $r = 8$ and scaling factor $\alpha = 32$ for fine-tuning. Table 3.1 summarizes the configurations explored.

The training configuration was consistent across AMI and AliMeeting. For NOTSOFAR-1, we focused on enhancing the ASR back-end of our challenge-trained system rather than training new systems. All models were trained on a single A100-80GB GPU.

Our separation and diarization pipelines use speaker embeddings extracted by the ECAPA-TDNN model from Speechbrain [71]. Hyperparameter optimization

involves a two-step process. First, we determine the optimal segmentation threshold and ASR collar (padding for leakage removal) using Oracle clustering. Second, we optimize clustering parameters (threshold and minimum cluster size) based on this threshold. Optimization is performed with respect to DER or cpWER, depending on the task.

3.4 Results

3.4.1 Effect of speaker embeddings as inputs

Table 3.2 reports DER and cpWER on the development sets for pipelines extracting embeddings from mixtures versus separated sources, as detailed in Section 3.2.6. Hereafter, we refer to embeddings from the original mixture as *mixture embeddings* and those from separated signals as *source embeddings*. For AliMeeting and AMI, we use the original ToTaToNet systems (WavLM and DPRNN). For NOTSOFAR-1, we use our best challenge system. To expedite optimization, we use the Whisper *small.en* and *small* models for ASR.

We observe a general trend that datasets with higher overlap tend to benefit from using source embeddings. On AliMeeting, SA-ASR performance improves, but diarization degrades with source embeddings. Conversely, on NOTSOFAR-1, the opposite occurs. The benefit of source embeddings for diarization was further confirmed in the DISPLACE 2024 challenge, where this technique was part of the winning system. Further analysis revealed that merging clusters to improve diarization can sometimes introduce artifacts into the separated signal that cause ASR hallucinations, worsening the tcpWER. This highlights a limitation of using SA-ASR to evaluate separation performance.

The fact that source embeddings can improve performance with an off-the-shelf model is promising. As shown in Section 3.4.4, fine-tuning the ASR system on separated sources yields significant SA-ASR improvements, suggesting that speaker embedding models could also benefit from fine-tuning on such data. For the remainder of the experiments, we adopt the method that yields the best development performance for each task and dataset.

3.4.2 Performance of different ToTaToNet architectures

This section evaluates the performance of the proposed ToTaToNet architectures on the AMI and AliMeeting datasets. Table 3.3 presents SA-ASR results on AMI using PixIT-based separation and Whisper *large-v3*. MossFormer2 outperforms DPRNN for both SSL models, aligning with findings for supervised speech separation [74]. An unexpected observation is the underperformance of W2v-BERT SSL features compared to WavLM; this is analyzed further in Section 3.4.3.

To assess broader applicability, we also trained the best-performing architecture

Table 3.2: DER (%) and cpWER (%) for different embedding extraction methods across datasets on the development split.

Dataset	Overlap (%)	Embeddings input	DER (%)				cpWER (%)			
			FA	MD	SC	total	sub	del	ins	total
AMI	14.6	Mixtures	4.9	6.3	4.8	**16.0**	7.3	20.0	2.3	**29.6**
		Sources	4.9	6.3	8.5	**19.7**	6.7	21.7	2.3	**30.7**
AliMeeting	20.4	Mixtures	4.6	6.6	6.1	**17.4**	16.2	22.2	3.4	**41.8**
		Sources	4.6	6.6	7.1	**18.3**	15.6	22.2	3.2	**41.0**
NOTSOFAR-1	39.4	Mixtures	4.2	9.1	9.4	**22.7**	8.4	22.2	4.2	**34.9**
		Sources	4.2	9.1	8.1	**21.3**	8.2	23.1	4.5	**35.8**

(WavLM with MossFormer2) on AliMeeting, with the results shown in Table 3.4. These findings confirm the generalizability of the trends. The PixIT-based systems outperform the CSS-based NOTSOFAR-1 baseline in all configurations.

Table 3.5 provides the diarization results. Performance remains consistent across configurations, suggesting that while more capable masking networks enhance separation, this does not directly translate to better diarization with the current, relatively simple diarization module. A trade-off exists between performance and architectural size in the masking networks (see Table 3.1). While MossFormer2 outperforms DPRNN, the improvement is slight, questioning the utility of the increased computational cost.

Table 3.3: tcpWER (%) and cpWER (%) for various ToTaToNet architectures with speaker attribution via diarization or separation on the AMI-SDM dataset using Whisper large-v3.

SSL model	Masking network	Speaker attribution	Attribution model	cpWER (%)				tcpWER (%)			
				sub	del	ins	total	sub	del	ins	total
Not used	Not used	Diarization	pyannote 3.1	7.2	27.8	4.8	**39.7**	6.1	29.5	6.4	**42.0**
Not used	Conformer	Diarization	NeMo	10.7	19.2	7.0	**36.9**	10.6	20.4	8.7	**39.7**
WavLM	DPRNN	Diarization	ToTaToNet	7.5	26.0	3.4	**36.9**	6.4	27.8	5.4	**39.5**
		Separation	ToTaToNet	7.0	19.6	2.8	**29.3**	7.3	21.4	4.6	**33.4**
WavLM	MossFormer2	Diarization	ToTaToNet	7.3	26.3	3.3	**36.9**	6.2	28.0	5.0	**39.2**
		Separation	ToTaToNet	6.9	19.4	2.6	**28.9**	7.1	21.3	4.5	**32.9**
W2v-BERT	DPRNN	Diarization	ToTaToNet	7.3	26.5	3.3	**37.1**	6.2	28.2	5.1	**39.4**
		Separation	ToTaToNet	7.3	22.7	2.6	**32.6**	7.7	23.9	4.5	**36.0**
W2v-BERT	MossFormer2	Diarization	ToTaToNet	7.5	26.2	3.2	**36.8**	6.2	28.0	5.0	**39.2**
		Separation	ToTaToNet	7.8	19.6	3.2	**30.6**	7.4	21.8	5.4	**34.7**

3.4.3 Comparison of SSL features

Surprisingly, the results in Tables 3.3 and 3.5 show that W2v-BERT did not improve over WavLM, and in some cases, led to performance degradation. To investigate this, we trained separate segmentation and separation models.

For diarization, we trained a segmentation model on AMI using the same

Table 3.4: tcpCER (%) and cpCER (%) for various ToTaToNet architectures with speaker attribution on the AliMeeting channel 1 dataset using Whisper large-v3.

SSL model	Masking network	Speaker attribution	Attribution model	cpCER (%)				tcpCER (%)			
				sub	del	ins	total	sub	del	ins	total
Not used	Not used	Diarization	pyannote 3.1	17.3	38.5	10.0	**65.9**	9.8	46.0	17.5	**73.3**
Not used	Conformer	Diarization	NeMo	15.6	27.0	7.0	**49.5**	15.2	28.0	8.1	**51.3**
WavLM	DPRNN	Diarization	ToTaToNet	18.3	37.6	9.2	**65.1**	10.1	45.6	17.2	**72.9**
		Separation	ToTaToNet	10.8	28.6	2.7	**42.1**	13.4	30.9	5.0	**49.4**
WavLM	MossFormer2	Diarization	ToTaToNet	19.1	37.0	8.5	**64.6**	10.3	45.5	17.0	**72.8**
		Separation	ToTaToNet	12.4	25.0	3.3	**40.7**	15.1	28.8	7.1	**51.1**

Table 3.5: DER (%) on AMI-SDM and AliMeeting channel 1 for different ToTaToNet systems. State-of-the-art as of December 2024 is denoted with 🏆.

SSL model	Masking network	λ	DER (%)			
			FA	MD	SC	total
AMI-SDM systems						
Han et al. [82]						**15.4** 🏆
WavLM	DPRNN	1.0	4.4	7.2	5.5	**17.1**
	DPRNN	0.5	3.9	8.2	5.6	**17.7**
WavLM	MossFormer2	0.5	5.0	8.5	3.9	**17.5**
W2v-BERT	DPRNN	0.5	5.0	8.8	3.9	**17.6**
	MossFormer2	0.5	4.9	8.6	4.2	**17.7**
AliMeeting systems						
Härkönen et al. [83]						**13.2** 🏆
WavLM	DPRNN	1.0	4.7	6.5	8.3	**19.5**
	DPRNN	0.5	5.8	7.3	8.3	**21.4**
WavLM	MossFormer2	0.5	6.8	6.9	7.7	**21.4**

hyperparameters as in PixIT. Audio representations were extracted directly from the SSL model and passed through 4 LSTM layers and a final linear layer to predict speaker activity. For separation, we used the same DPRNN masking network and TasNet encoder hyperparameters as in the PixIT models. We evaluated these models on the WSJ0-2Mix dataset to directly measure separation gains (SDR, SDRi, SI-SDRi) on artificial mixtures where ground truth sources are available. In both cases, W2v-BERT and WavLM were fine-tuned under the same conditions as their respective ToTaToNet models.

Table 3.6 shows a clear benefit from using W2v-BERT when the tasks are trained disjointly. For diarization, a relative 10% improvement in DER is observed. For separation, we see a 14% relative improvement in dB across all metrics. These results confirm that W2v-BERT should improve performance for both tasks, which was not the case in the end-to-end ToTaToNet models.

Table 3.6: DERs (%) for segmentation models trained on AMI-SDM and separation gains (dB) for models trained on WSJ0-2Mix. Both models use a DPRNN masking network.

SSL model	DER (%)				Gains (dB)		
	FA	MD	SC	total	SDR	SDRi	SI-SDRi
WavLM	4.3	8.8	6.2	**19.2**	16.4	16.2	16.0
W2v-BERT	4.0	7.2	5.8	**17.3**	18.6	18.4	18.2

This suggests a potential bottleneck in how the ToTaToNet models leverage SSL representations for both tasks simultaneously. To investigate, we performed a layer-wise analysis by freezing the SSL models and training a weighted average of their 24 transformer layers for each task.

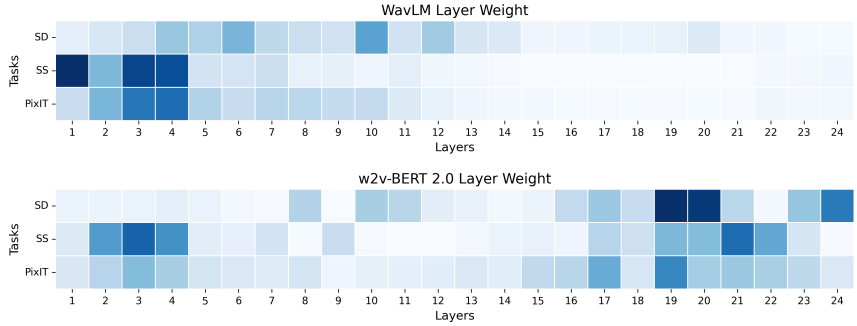

Figure 3.8: Layer contribution of W2v-BERT 2.0 and WavLM (large version) for Speaker Diarization (SD), Speech Separation (SS), and the joint task (PixIT).

As shown in Figure 3.8, for WavLM, both diarization and separation tasks activate the early layers (1-7), which are known to be important for speaker identity. This behavior translates well to PixIT, which also activates these layers. In contrast, for W2v-BERT, diarization activates the top layers while separation activates both early and top layers. This discrepancy forces the joint PixIT model to compromise, leveraging a mix of early and top layers, which may trade off performance in each task. This suggests that for a joint model like PixIT, using an SSL model where optimal representations for each task reside in different layers presents a challenge. Further investigation is needed to explore more effective ways to integrate features from such SSL models.

3.4.4 Fine-tuning ASR

Fine-tuning ASR models on in-domain data typically yields significant improvements. We investigated this by creating two fine-tuned versions of Whisper *large-v3*: one trained on the original AMI-SDM training data with merged transcripts, and another trained on separated audio sources and their corresponding speaker-attributed

Table 3.7: cpWER (%) and tcpWER (%) on AMI-SDM test set for various fine-tuned Whisper large-v3 models and attribution methods, with relative changes compared to no fine-tuning.

ASR fine-tuning	Attribution	cpWER (%)				tcpWER (%)				Relat. change (%)	
		sub	del	ins	total	sub	del	ins	total	cpWER	tcpWER
None	Diarization	7.3	26.4	3.2	**36.9**	6.4	27.9	4.9	**39.2**	–	–
On original audio	Diarization	9.9	14.0	7.1	**30.9**	8.2	15.9	9.3	**33.4**	-16.2	-14.8
On separated sources	Diarization	7.8	22.9	2.3	**32.9**	7.1	24.3	3.8	**35.1**	-10.8	-10.5
None	Separation	5.8	21.7	1.7	**29.3**	6.5	22.8	2.8	**32.2**	–	–
On original audio	Separation	14.1	9.5	19.1	**42.8**	11.1	13.1	23.1	**47.3**	+46.1	+47.0
On separated sources	Separation	4.1	16.7	1.8	**22.6**	6.8	14.4	3.7	**24.8**	-22.9	-23.0

transcripts. Training utterances were created from the word-level transcriptions provided with the AMI dataset.

Both models were trained for three epochs using identical hyperparameters. Table 3.7 presents the results on the AMI-SDM test set. When the original single-channel audio is used as input, both fine-tuned models improve over the base model, with the model fine-tuned on single-channel audio performing best.

However, when these models are applied to each separated source independently, the results diverge. The model fine-tuned on multi-speaker audio shows a noticeable decline in accuracy. Conversely, the model fine-tuned on separated sources delivers a substantial improvement. This result demonstrates that when using PixIT for separation, it is crucial to fine-tune the ASR model on separated audio that matches the test-time input.

3.4.5 Improving on our NOTSOFAR-1 Challenge submission

Table 3.8 presents results for our NOTSOFAR-1 Challenge systems on the eval-small dataset, using the same ToTaToNet checkpoint as our challenge submission. Similar to the AMI experiments, fine-tuning the *large-v3* model on single-channel audio results in a large WER increase, likely due to frequent hallucinations.

Conversely, fine-tuning on separated sources significantly improves the tcpWER to 33.7%, a 20% relative reduction compared to the baseline, again demonstrating the effectiveness of aligning fine-tuning with the test configuration. Notably, our method slightly improves on the NOTSOFAR-1 baseline when using an identical downstream ASR model (Whisper *large-v2*). This shows that PixIT is a promising alternative to CSS, even when domain-matched synthetic data is available for the CSS system.

3.4.6 Effect of the timestamp fix heuristic

The effect of our timestamp correction heuristic, introduced in Section 3.2.7, is detailed in Table 3.9. The heuristic generally mitigates error increases caused by the time collar, with the most pronounced improvements seen in the model fine-tuned

Table 3.8: Performance on the NOTSOFAR-1 eval-small split. This table includes cpWER, tcpWER, and the relative tcpWER change with respect to the baseline system.

System	cpWER (%)				tcpWER (%)				ΔtcpWER (%) (relative)
	sub	del	ins	total	sub	del	ins	total	
NOTSOFAR-1 baseline	11.3	22.0	7.4	40.7	10.0	23.3	8.8	42.1	0.0
Our NOTSOFAR-1 submission	10.7	14.2	9.8	34.7	10.3	17.6	13.2	41.1	-2.4
large-v2	7.7	25.4	3.7	36.8	7.4	27.9	6.3	41.7	-1.0
large-v3	7.1	24.9	3.7	35.6	7.2	27.5	6.3	40.9	-2.9
large-v3, ft. on single channel	21.8	14.0	45.0	80.8	14.2	21.4	52.4	88.1	+109.3
large-v3, ft. on sep. sources	8.0	16.3	6.0	30.3	7.2	18.3	8.1	33.7	-20.0

on separated sources. These results demonstrate that while PixIT can introduce timestamp errors due to long silences in the separated sources, such errors can be largely corrected with lightweight post-processing.

Table 3.9: Effect of the timestamp fix heuristic on the tcpWER metric. This table presents the total cpWER, tcpWER before and after the fix, the relative error proportion from the collar, and the relative change in collar errors after fixing.

System	cpWER (%)	tcpWER (%)		Rel. collar err. proportion (%)	Rel. change to collar err. from fix (%)
		before fix	after fix		
large-v3	35.6	41.6	40.9	13.2	-11.7
large-v3, ft. on sep. sources	30.3	34.8	33.7	12.9	-24.4

3.5 Conclusion

This chapter presented and evaluated PixIT, a framework for jointly training speaker diarization and unsupervised speech separation using only diarization labels from real-world data. The experiments investigated its effectiveness and the influence of different architectural choices and downstream components. The results indicated that advanced masking networks like MossFormer2 can improve separation within the ToTaToNet architecture. For self-supervised features, we found that it is important for the layer contributions for the separation and diarization tasks to be similar.

The results demonstrate that fine-tuning an ASR system on PixIT-separated sources significantly boosts downstream SA-ASR performance. Notably, these gains are greater than those for a standard diarization-based SA-ASR system where the ASR is fine-tuned on the original mixtures.

PixIT-separated sources also show potential for speaker embedding extraction, contributing to the winning submission in the DISPLACE 2024 challenge. This suggests that fine-tuning speaker embedding models on separated sources is a promising direction for future work.

PixIT is a strong competitor to traditional SA-ASR methods, outperforming a CSS baseline on the NOTSOFAR-1 Challenge dataset without relying on the

tailored synthetic data used by the CSS system. Therefore, in addition to being easier to train, PixIT can rival supervised separation approaches on real-world mixtures.

Chapter 4

Diarization-guided multi-speaker embeddings

This chapter is based on the work in publication V.

Building on the theme of domain mismatch, this chapter addresses some of the challenges faced by speaker embedding systems in multi-speaker scenarios. As noted in Section 1.2.2, embeddings trained on clean, single-speaker data degrade significantly when processing real-world audio containing overlapping speech. Furthermore, current systems process speakers sequentially, which is inefficient. This chapter presents a method to extract robust and discriminative embeddings for all speakers concurrently from a single audio chunk, using guidance from a diarization model to improve both accuracy and processing speed.

4.1 Background

High-quality speaker embeddings are essential for multi-speaker speech processing tasks. In speaker diarization, EEND-vector clustering (EEND-VC) systems rely on speaker embeddings derived from local segmentation outputs to cluster and stitch these local segments [16]. Similarly, in multi-speaker automatic speech recognition (ASR), transducer-based systems produce segment-wise transcriptions with timestamp estimates, which are subsequently attributed to individual speakers using speaker embeddings [84]. For voice conversion of long-form audio, it is desirable that speaker embeddings are modelled consistently [85]. In all the above use cases, it would be beneficial for all the speakers to be modeled concurrently.

Previous studies have explored joint training of ASR and segmentation models with multi-speaker embeddings [17, 84], but these approaches have underperformed compared to standalone embedding systems [86]. This discrepancy is likely due to the difference in data quality: speaker verification datasets, which are more easily annotated, tend to be larger and more diverse than those used for ASR and

speaker diarization [37]. These datasets contain only single-speaker utterances, leading to a domain mismatch when applied in multi-speaker scenarios. To address this, guided speaker embeddings (GSE) were recently introduced for multi-speaker environments [28]. GSE is trained on synthetic multi-speaker mixtures derived from a speaker verification dataset, with oracle activity labels guiding the process. Activity labels for both target and interference speakers are used as additional inputs to the embedding encoder and for masking attention scores. However, these systems still produce embeddings for only one speaker at a time.

This chapter proposes extending GSE by modeling all speakers present in a chunk at once. Additionally, since the practical deployment of a GSE system relies on a speaker segmentation model, we also propose using its output as a guide for training instead of oracle labels. Features from a segmentation model can offer more detailed guidance. For example, areas of high confidence indicate it is easier to extract speaker-specific information there.

The main contributions of this chapter are as follows:

- Proposing a diarization-guided training method for multi-speaker embedding systems.
- Introducing a modified attention module to allow for multi-speaker modeling in existing speaker embedding models.
- Proposing a new validation metric optimized for speaker embeddings in a multi-speaker context.
- Providing a thorough evaluation of the multi-speaker embeddings on multiple speaker diarization and verification datasets.
- Open-sourcing the code for the above[1].

4.2 Method

Figure 4.1 illustrates our joint architecture, which combines a local speaker segmentation model with a speaker embedding model using a shared feature extractor. We opt for an SSL-based features extractor, namely WavLM, since it demonstrates good performance in both speaker diarization and speaker verification tasks [68]. It is also the choice for state-of-the-art for speaker diarization as of writing [82, 87]. We use the same LSTM-based segmentation probing head as in [87]. For the embedding module, we use an ECAPA-TDNN, which has been shown to perform better than smaller probing heads [25, 88].

Given the frame-level features extracted from an audio chunk $\mathbf{x} \in \mathbb{R}^{T \times F}$ and assuming a maximum of $K_{\max}$ speakers, the segmentation module extracts powerset

[1] https://github.com/joonaskalda/multi-speaker-embeddings

features $\mathbf{p} \in \mathbb{R}^{T \times K_{\text{ps}}}$, where K_{ps} is the number of powerset classes. These are binarized and converted into a multi-label format $\mathbf{a} \in \{0,1\}^{T \times K_{\max}}$ [89]. The powerset features are concatenated with $\mathbf{x}$ to form combined features of dimension $F + K_{\text{ps}}$, which are fed into the embedding encoder. Only the input channel dimension of the encoder is modified in our approach.

The encoder output $\mathbf{h} \in \mathbb{R}^{T \times D}$ is reshaped to introduce a speaker dimension, resulting in $\mathbf{h'} \in \mathbb{R}^{K_{\max} \times T \times (D/K_{\max})}$. The attention module remains unchanged from the original ECAPA-TDNN, except that all the channel dimensions are scaled down by a factor of $K_{\max}$, except for the bottleneck attention dimension, which is kept at 128. The batch size after the encoder is effectively increased $K_{\max}$ times, with the speakers being processed in parallel.

Similarly to GSE, we apply silent masking for each predicted speaker but use binarized predicted speaker activations instead of oracle labels. The embedding dimension for the predicted multi-speaker embeddings $\{\mathbf{e_1}, \ldots, \mathbf{e_{K_{\max}}}\} \in \mathbb{R}^{192}$ is kept unchanged from the original ECAPA-TDNN.

Note that this approach would require slight modifications if the encoder output channel dimension D is not divisible by $K_{\max}$ by e.g. adding an adaptation layer. In the above we also assumed, for simplicity, that the embedding encoder leaves the temporal resolution unchanged, as is the case for ECAPA-TDNN, used in our experiments. If that is not the case, the speaker activation masks should be interpolated to match the embedding output temporal resolution.

To train the multi-speaker embedding model we use synthetic VoxCeleb mixtures as in [28]. We use the standard ArcFace loss [27] but only compute it for an embedding if the segmentation model correctly predicts the corresponding speaker's activation for at least one second.

In our experiments, we use a segmentation probing head trained using powerset loss [89], but this is not a requirement. The only new components that need to be trained in our method are in the speaker embedding branch. For the segmentation branch and feature extractor, any off-the-shelf model can be utilized, and no specific adaptation is needed. Our proposed training method generalizes naturally to any local segmentation and speaker embedding architecture, with no requirement for a shared feature extractor.

To summarize, our method builds on top of GSE by

- Changing the attention module to extract multi-speaker embeddings instead of single-speaker embeddings.
- Utilizing detailed information from the segmentation module.

4.2.1 Validation metric

The standard validation metric for speaker embedding models is the equal error rate (EER), computed on single-speaker utterance trials. However, this does not

reflect performance in multi-speaker scenarios. We argue that speaker diarization performance is a more appropriate metric, as the clustering stage directly depends on embedding quality. We therefore propose using diarization performance for both evaluation and validation. Validation is challenging; normally, it would require hyperparameter optimization after each epoch, which is a resource-intensive process.

To address this, we propose a simplified pipeline for validation (Figure 4.2). In each validation batch, all audio chunks are sampled from the same file using a sliding window, ensuring that the first chunk starts at the beginning of the audio and the last one stops at the file's end. The step size S between consecutive chunks is chosen so that all chunks are evenly spaced i.e. $S = \frac{D-T}{B-1}$, where D is the file duration, B is the batch size, and T is the chunk length. Chunks overlap if and only if $BT > D$.

The batch is then fed to the model, which returns segmentation predictions and speaker embeddings for each chunk. These embeddings are then clustered using the K-means algorithm, where the number of clusters is fixed based on the oracle number of speakers in the file used to create the batch. For efficiency, we assume the number of speakers is known, which eliminates the need for tuning clustering parameters. Finally, segments are assigned to speakers as in a standard diarization pipeline, and a diarization error rate (DER) over the validation files is calculated based on the pipeline output and the corresponding cropped reference.

4.3 Experiments

4.3.1 Datasets

The feature extractor and diarization branch are trained on a composite dataset consisting of AMI-SDM [76], AliMeeting (first channel) [77], AISHELL-4 (first channel) [90], MSDWILD [91], RAMC [92], and EGO4D [93]. Since EGO4D does not include an evaluation set, we use it only for training and validation.

The speaker embedding systems are trained on either VoxCeleb 1 and 2 utterances [37, 94] or synthetic mixtures generated from these datasets.

4.3.2 Data simulation

For training speaker embedding systems in multi-speaker contexts, we use 10-second synthetic VoxCeleb mixtures, following the approach in GSE. However, we modify the simulation method to better reflect real-world multi-speaker scenarios. Specifically, we allow arbitrary speaker order and permit delays of up to 0.5 seconds after the preceding utterance ends to introduce natural silent regions. Utterance lengths are sampled from an exponential distribution with $\lambda = 0.2$, truncated to the range [1,10] seconds. Additionally, we apply room background noise to the mixtures using data from [55] and simulated room impulse responses from [56].

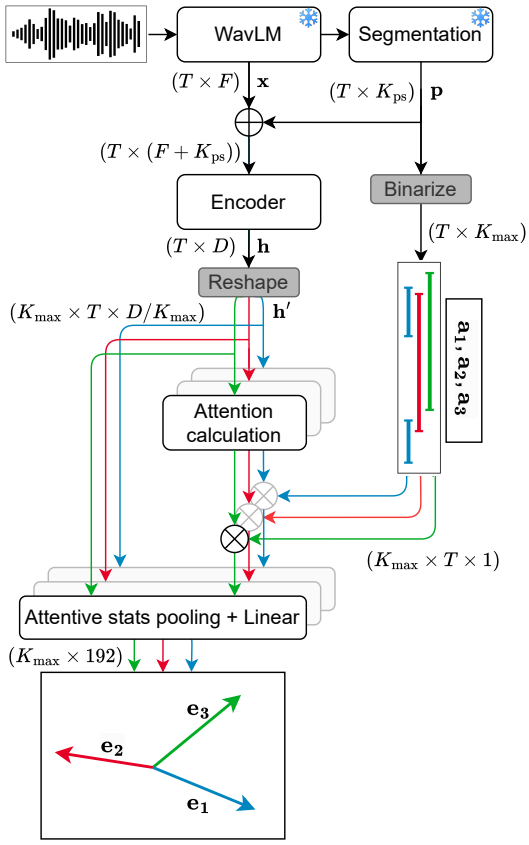

Figure 4.1: Proposed joint architecture for a maximum of $K_{\text{max}} = 3$ speakers per audio chunk. We opt for a segmentation branch trained using a powerset loss, but this is not a requirement.

4.3.3 Implementation details

Segmentation model. Training chunks are 10 seconds long, with a maximum of $K_{\text{max}} = 3$ speakers per chunk. We train a standard diarization system using powerset loss, assuming that no more than two speakers are active at a time, resulting in $K_{\text{ps}} = 7$ powerset classes, assuming no more than two concurrent speakers. Our segmentation module follows the architecture from [87].

We use a WavLM Base+ model as the shared feature extractor, fine-tuned together with the segmentation module as in [82]. The learning rates are set to 10^{-5} for WavLM and 3×10^{-4} for the segmentation module, with a batch size of 32. The embedding and segmentation models use separate learnable weighted sums of the WavLM layers.

Speaker embedding model. Our speaker embedding extractor is an ECAPA-TDNN model with 1024 channels. As a baseline, we train an unguided single-embedding system on 3-second utterances with a batch size of 512.

For all other speaker embedding systems, we adopt the training strategy from

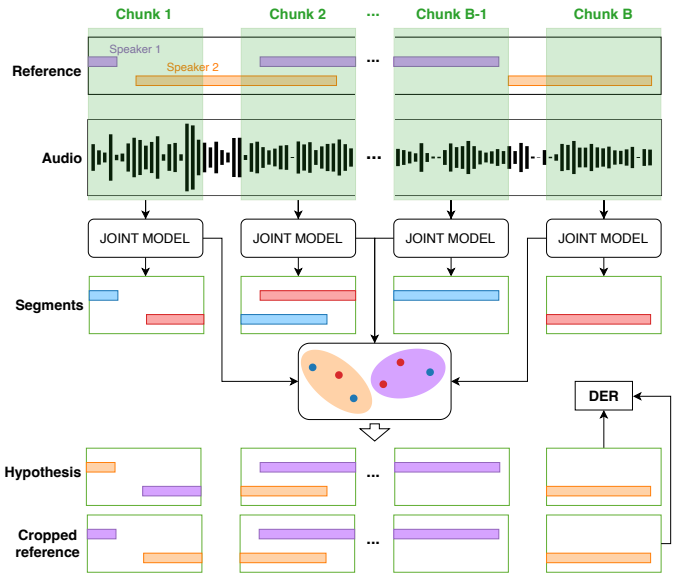

Figure 4.2: Proposed validation pipeline with a batch size of B.

[28]. We employ the Adam optimizer with a cyclical learning rate schedule over three cycles, using a batch size of 128 mixtures. This results in an effective batch size of 384 for ArcFace loss computation. Each cycle consists of 50k steps, beginning with a 1k-step warm-up phase, followed by cosine annealing decay. The learning rate starts at a peak of 10^{-3} and decays by a factor of 0.75 at the start of each new cycle.

Validation metrics. For our proposed DER-based validation metric, we randomly sample 10 files from each dataset's validation set, yielding a total of 58 batches [2]. The baseline validation metric is the equal error rate (EER), calculated on the widely used VoxCeleb test set 1-O, containing 37611 test trials based on single-speaker utterances [37].

Speaker diarization inference. For speaker diarization inference, we use the pyannote 3.1 pipeline [95] with the same configuration as [28]. After selecting the optimal checkpoint based on the validation metric, we optimize the speaker diarization clustering hyperparameters for each system using Optuna [96]. Hyperparameter tuning is performed on the compound validation set using the multivariate Tree-structured Parzen Estimator for 100 iterations.

Evaluation. Direct evaluation of speaker embeddings in a multi-talker context would require a multi-talker real-world verification dataset, which currently does not exist. Previous work has used synthetic mixtures based on VoxCeleb to assess multi-speaker performance [28], but these do not accurately capture real-world conversational dynamics [97]. Because of this, we assess embedding quality indirectly

[2] AliMeeting validation set only has 8 files

	Training guide	Params	Validation Metric	EER (%)	AISHELL4	AMI-SDM	AliMeeting	MSDWILD	RAMC	Macro-avg.
Oracle clustering	-	-	-	-	1.3	1.9	2.3	2.8	3.5	2.5
Single-embedding	Unguided	24.3M	EER	1.1	4.7	8.3	10.2	12.6	8.1	8.4
			DER	1.4	5.1	8.0	8.3	12.2	7.9	8.1
Single-embedding [28]	Oracle	24.3M	EER	1.8	2.9	3.8	3.6	12.1	7.3	5.9
			DER	1.9	2.7	3.7	3.6	11.8	7.2	5.9
Multi-embedding	Oracle	22.5M	EER	1.6	3.0	6.2	4.2	13.4	6.7	6.5
			DER	2.2	3.2	5.6	4.4	11.2	7.1	6.2
Multi-embedding	Diarization	22.5M	EER	1.7	3.2	3.7	5.0	11.4	7.3	6.2
			DER	1.8	2.5	3.8	3.5	11.4	7.2	5.7

Table 4.1: Comparison of single-speaker (single-embedding) and multi-speaker (multi-embedding) embedding systems with different guiding mechanisms and validation metrics. We report EER on VoxCeleb 1-O and speaker confusion rates on diarization datasets, as well as the macro-average (Macro-avg) for the latter. Speaker confusion using oracle clustering is included as a topline reference.

via diarization pipeline performance. With a fixed local segmentation model, false alarm and missed detection rates are constant. Consequently, we only report speaker confusion rates, determined by clustering and directly reflecting embedding quality. Scores on the VoxCeleb test set 1-O are also reported for reference.

4.4 Results

A comparison of multi-speaker and single-speaker embeddings, along with different guiding methods, is presented in Table 4.1. All guided systems outperform the standard unguided system in diarization but underperform in EER, consistent with the findings of [28]. Switching from oracle-guided single-embedding to multi-embedding leads to a performance drop, which is expected since the encoder must now model all participating speakers rather than a single target speaker, while the model size is slightly reduced (due to scaling down the channel dimension in the attention module). However, this degradation is mitigated by replacing oracle-guided training with diarization-guided training.

Multi-speaker models are also more compact, as the channel dimension is scaled down by $K_{\max}$ after the encoder. The oracle clustering system serves as an upper bound, assuming perfect speaker clustering, with non-zero speaker confusion scores arising only from intra-chunk segmentation errors. Validation using the proposed simplified diarization pipeline demonstrates clear improvements in most cases, with comparable results for the GSE system, where the selected checkpoints had very similar performance.

Figure 4.3 shows the EER and speaker confusion scores as a function of step count for the diarization-guided multi-speaker embedding system. The two curves display low correlation after initial fast convergence, further highlighting that the standard VoxCeleb 1-O EER is not optimal in multi-speaker applications.

Dataset	MSE	ResNet34	SOTA
AISHELL-4	12.0	12.4	10.6 [87]
AMI-SDM	15.7	16.5	15.4 [82]
AliMeeting	15.9	17.4	13.2 [83]
MSDWILD	22.9	21.6	19.6 [87]
RAMC	11.8	11.1	11.1 [83]
Macro-average	15.0	15.1	13.4

Table 4.2: Comparison of DERs for multi-speaker embeddings (MSE), and ResNet34 embeddings across datasets. State-of-the-art (SOTA) DERs are provided for reference.

Dataset	Ovr. (%)	Spk. #	RTF Imp. (%)
AISHELL-4	5.0	2.0	39
AMI-SDM	14.6	2.2	43
AliMeeting	20.4	2.8	53
MSDWILD	12.4	2.0	40
RAMC	9.4	1.8	36
Macro-average	12.0	2.1	42

Table 4.3: Comparison of overlapping speech percentage (Ovr.), average speaker count per chunk (Spk. #), and relative RTF improvements (RTF Imp.) across datasets.

In Table 4.2, we compare our diarization-guided multi-embedding system to a state-of-the-art ResNet-based speaker embedding model [26] from pyannote 3.1 based on DER, keeping the segmentation model the same. State-of-the-art DER scores are also provided for reference. Although the ResNet system employs a more sophisticated training strategy, including speed augmentation and large-margin fine-tuning, our system displays competitive results across the board, with significantly better results on the higher-overlap datasets AMI-SDM and AliMeeting.

Table 4.3 explores the real-time factor (RTF) of multi-speaker embeddings compared to single-speaker embeddings, which requires encoding each speaker in a chunk separately. We first measure the total time for inference using a diarization

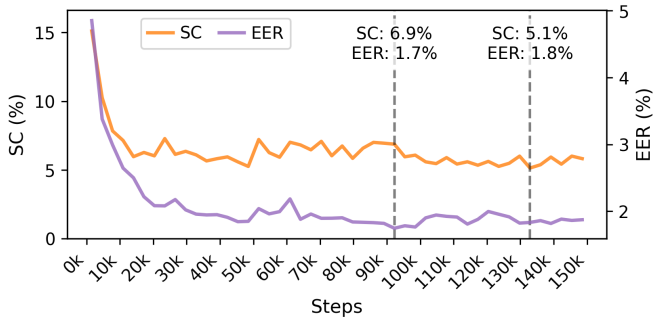

Figure 4.3: Validation EER and speaker confusion (SC) (%) scores as a function of step count. The optimal checkpoints based on either metric are highlighted with a dotted line.

pipeline with oracle speaker clustering, where no speaker embeddings are calculated. Then we measure the increase in RTF from performing clustering using either system. Comparing the results for the two systems gives us the relative decrease in RTF. We also report both the percentage of frames containing overlapped speech and the average number of speakers in a 10-second chunk sampled from the dataset. The latter directly reflects the number of separate forward passes required by the single-speaker embedding encoder, in contrast to the single pass needed for our multi-speaker approach. Even for the relatively low-overlap scenarios represented by AISHELL-4 and RAMC, the multi-embedding system achieves an RTF relative improvement of at least 36%.

4.4.1 Future work

Training of our diarization-guided multi-speaker embeddings relies on synthetically generated mixtures, which, while useful, fail to capture the complexity of real-world conversation dynamics [97]. Training or fine-tuning the embeddings directly on real-world data should help performance, although a comparatively small number of speakers in real-world conversational datasets poses a challenge here.

We keep the speaker embedding encoder unchanged from the single-speaker case, but since it now has to model multiple speakers, the architecture should be optimized for this.

4.5 Conclusion

This chapter introduced a novel diarization-guided training method for multi-speaker embeddings. We extended guided speaker embeddings by modeling speakers concurrently using diarization-based guidance. We also introduced a novel clustering-based validation metric for training embeddings in a multi-speaker context, which we showed to be more effective than standard speaker verification EER based on single-speaker utterances. Keeping the embedding encoder unchanged, we compared the effects of both modifications on multiple speaker diarization datasets. We showed that while switching to modeling multiple speakers concurrently degrades performance, this deficit is offset by using diarization-based guidance, which contains more information and better matches testing conditions. The result is a speaker embedding system that is smaller, more accurate, and considerably faster than comparable systems trained using previous methods.

Chapter 5

Conclusion

This thesis has examined some of the central challenges in multi-talker speech processing (MTSP), particularly those arising from the mismatch between controlled training conditions and the complexity of real-world acoustic scenes. The work explored how the limitations of current systems—such as annotation ambiguities, reliance on synthetic training data, and difficulties in handling overlapping speech—can be mitigated by incorporating more realistic data constraints and evaluation-oriented objectives into the training process.

The contributions of this thesis focus on three related areas. First, a collar-aware loss function for speaker change detection was proposed, designed to better reflect the temporal tolerance of manual annotations and evaluation protocols. This approach improved detection accuracy and produced models more suitable for streaming applications. Second, a joint training framework called PixIT was introduced to learn speaker diarization and speech separation simultaneously from multi-speaker recordings. By combining permutation-invariant training for diarization with mixture-invariant training for separation, PixIT enables long-form speech separation to be trained on real-world data. Third, a diarization-guided method for learning multi-speaker embeddings was developed. By using guidance from a diarization model, modifying the embedding encoder, and adopting a more suitable validation metric, this approach produced embeddings that were more accurate, smaller, and faster to compute in multi-speaker conditions.

Taken together, these contributions aim to move beyond idealized assumptions toward a more integrated system design. By considering annotation ambiguities, emphasizing conversational data, and combining related tasks, this thesis seeks to provide more practical and robust approaches to multi-speaker processing. It is hoped that the methods described here will prove useful for the development of future MTSP systems.

5.1 Future work

The research presented in this thesis opens up several promising avenues for future investigation. The core principles of aligning training with evaluation and pursuing joint learning paradigms can be extended to address some of the remaining challenges in the field.

A natural extension of collar-aware training is its application to speaker diarization. End-to-end diarization systems typically use a frame-wise binary cross-entropy loss. This is the same loss that is used in speaker change detection and suffers from a similar mismatch with the standard evaluation metric. The diarization error rate (DER) has often been computed with a forgiveness collar, though recent trends are moving away from this practice, as the collar can mask performance in the most challenging regions. Nevertheless, the inherent ambiguity in human segment annotations, exacerbated by differences in annotation guidelines, suggests that collar-aware training for diarization remains a promising direction. Such an approach could give models greater flexibility to learn from annotations of varying quality, potentially leading to substantial improvements in the temporal accuracy of end-to-end diarization systems.

Furthermore, this thesis motivates a deeper investigation into joint training paradigms that leverage the fact that different MTSP tasks extract speaker-specific information at varying resolutions. While PixIT successfully integrated diarization (time-level information) and separation (time-frequency-level information), a more comprehensive framework could also incorporate speaker embedding extraction (utterance-level information). Future research could focus on designing a unified architecture that jointly performs diarization, separation, and speaker identification. In such a system, information could flow bidirectionally between components: robust speaker embeddings could guide the separation module to better isolate a target speaker's voice, while cleaner separated signals could, in turn, be used to refine the speaker embeddings. This suggests an iterative refinement process, leading to a feedback cycle where improvements in one task directly enhance the performance of others within a single, powerful model. Training such a system on real-world data with a PixIT-style objective could lead to further improvements in accuracy and a more complete understanding of complex conversational dynamics.

Future work on multi-speaker embeddings could focus on moving beyond the limitations of synthetic data by using real-world conversational datasets for either training or fine-tuning the embeddings. Furthermore, as text-to-speech (TTS) technology improves, especially in its capacity to generate realistic conversational speech, the performance of embeddings trained on such enhanced synthetic data is also poised for improvement. This approach has shown promise for conversational speech recognition [98], although generating overlapping speech remains a challenge. Another promising avenue is the exploration of a joint training paradigm for the segmentation and embedding modules. While our initial experiments in this direction

were not fruitful, it is possible that an optimal configuration was missed within the high-dimensional hyperparameter space. Lastly, the embedding encoder architecture itself, which was adapted from single-speaker models, could be further modified and optimized specifically for the more complex task of concurrently modeling multiple speakers.

Bibliography

[1] Joonas Kalda and Tanel Alumäe. Collar-aware training for streaming speaker change detection in broadcast speech. In *Odyssey 2022: The Speaker and Language Recognition Workshop*, pages 141–147, 2022.

[2] Joonas Kalda, Clément Pagés, Ricard Marxer, Tanel Alumäe, and Hervé Bredin. PixIT: Joint training of speaker diarization and speech separation from real-world multi-speaker recordings. In *Odyssey 2024: The Speaker and Language Recognition Workshop*, pages 115–122, 2024.

[3] Joonas Kalda, Tanel Alumäe, Martin Lebourdais, Hervé Bredin, Séverin Baroudi, and Ricard Marxer. TalTech-IRIT-LIS speaker and language diarization systems for DISPLACE 2024. In *25th Annual Conference of the International Speech Communication Association, Interspeech 2024*, pages 1635–1639, 2024.

[4] Joonas Kalda, Séverin Baroudi, Martin Lebourdais, Clément Pagés, Ricard Marxer, Tanel Alumäe, and Hervé Bredin. Design choices for PixIT-based speaker-attributed ASR: Team ToTaTo at the NOTSOFAR-1 challenge. *Computer Speech & Language*, page 101824, 2026.

[5] Joonas Kalda, Clément Pagés, Tanel Alumäe, and Hervé Bredin. Diarization-Guided multi-speaker embeddings. In *26th Annual Conference of the International Speech Communication Association, Interspeech 2025*, pages 5233–5237, 2025.

[6] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al. State-of-the-art speech recognition with sequence-to-sequence models. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018*, pages 4774–4778, 2018.

[7] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning, ICML 2023*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518, 2023.

[8] Jon Barker, Shinji Watanabe, Emmanuel Vincent, and Jan Trmal. The fifth 'CHiME' speech separation and recognition challenge: dataset, task and baselines. In *19th Annual Conference of the International Speech Communication Association, Interspeech 2018*, pages 1561–1565, 2018.

[9] E Colin Cherry. Some experiments on the recognition of speech, with one and with two ears. *The Journal of the acoustical society of America*, 25(5):975–979, 1953.

[10] Xavier Anguera, Simon Bozonnet, Nicholas Evans, Corinne Fredouille, Gerald Friedland, and Oriol Vinyals. Speaker diarization: A review of recent research. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):356–370, 2012.

[11] Tae Jin Park, Kyu J Han, Manoj Kumar, and Shrikanth Narayanan. A review of speaker diarization: Recent advances with deep learning. *Computer Speech & Language*, 72:101317, 2022.

[12] Steve E. Tranter and Douglas A. Reynolds. An overview of automatic speaker diarization systems. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1557–1565, 2006.

[13] Yusuke Fujita, Naoyuki Kanda, Shota Horiguchi, Kenji Nagamatsu, and Shinji Watanabe. End-to-end neural speaker diarization with permutation-free objectives. In *20th Annual Conference of the International Speech Communication Association, Interspeech 2019*, pages 4300–4304, 2019.

[14] Federico Landini, Ján Profant, Mireia Díez, and Lukás Burget. Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: Theory, implementation and analysis on standard tasks. *Comput. Speech Lang.*, 71:101254, 2022.

[15] Shota Horiguchi, Yusuke Fujita, Paola Garcia, Shinji Watanabe, and Kenji Nagamatsu. EEND-EDA: End-to-end neural diarization with encoder-decoder based attractors. In *2022 IEEE Spoken Language Technology Workshop, SLT 2022*, pages 488–495, 2022.

[16] Keisuke Kinoshita, Marc Delcroix, and Naohiro Tawara. Advances in integration of end-to-end neural and clustering-based diarization for real conversational speech. In *22nd Annual Conference of the International Speech Communication Association, Interspeech 2021*, pages 3565–3569, 2021.

[17] Keisuke Kinoshita, Marc Delcroix, and Naohiro Tawara. Integrating end-to-end neural and clustering-based diarization: Getting the best of both worlds. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021*, pages 7198–7202, 2021.

[18] Hagai Aronowitz and Weizhong Zhu. Context and uncertainty modeling for online speaker change detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020*, pages 8379–8383, 2020.

[19] Ruiqing Yin, Hervé Bredin, and Claude Barras. Speaker change detection in broadcast TV using bidirectional long short-term memory networks. In *18th Annual Conference of the International Speech Communication Association, Interspeech 2017*, pages 1278–1282, 2017.

[20] Joseph P. Campbell. Speaker recognition: A tutorial. In *Proceedings of the IEEE*, volume 85:9, pages 1437–1462, 1997.

[21] Tomi Kinnunen and Haizhou Li. An overview of text-independent speaker recognition: From features to supervectors. *Speech Communication*, 52(1):12–40, 2010.

[22] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker identification and verification using gaussian mixture models. In *Speech Communication*, volume 32, pages 173–185, 2000.

[23] Najim Dehak, Patrick J. Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2011.

[24] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust DNN embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018*, pages 5329–5333, 2018.

[25] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification. In *21st Annual Conference of the International Speech Communication Association, Interspeech 2020*, pages 3830–3834, 2020.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pages 770–778, 2016.

[27] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2019*, pages 4690–4699, 2019.

[28] Shota Horiguchi, Tomoaki Moriya, Akitaka Ando, Tomo Ashihara, Hiroshi Sato, Naohiro Tawara, and Marc Delcroix. Guided speaker embedding. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2024*, pages 11276–11280, 2024.

[29] Yi Luo and Nima Mesgarani. TaSNet: Time-domain audio separation network for real-time, single-channel speech separation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018*, pages 696–700, 2018.

[30] Yi Luo, Zhuo Chen, and Takuya Yoshioka. Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020*, pages 46–50, 2020.

[31] Cem Subakan, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong. Attention Is All You Need in Speech Separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021*, pages 21–25, 2021.

[32] Zhong-Qiu Wang, Samuele Cornell, Shukjae Choi, Younglo Lee, Byeong-Yeol Kim, and Shinji Watanabe. TF-GRIDNET: making time-frequency domain models great again for monaural speaker separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2023*, pages 1–5, 2023.

[33] Sanyuan Chen, Yu Wu, Zhuo Chen, Jian Wu, Jinyu Li, Takuya Yoshioka, Chengyi Wang, Shujie Liu, and Ming Zhou. Continuous speech separation with conformer. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021*, pages 5749–5753, 2021.

[34] Aswin Sivaraman, Scott Wisdom, Hakan Erdogan, and John R. Hershey. Adapting speech separation to real-world meetings using mixture invariant training. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022*, pages 686–690, 2022.

[35] Scott Wisdom, Efthymios Tzinis, Hakan Erdogan, Ron J. Weiss, Kevin W. Wilson, and John R. Hershey. Unsupervised sound separation using mixture invariant training. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, pages 3846–3857, 2020.

[36] Jonathan G. Fiscus, Jerome Ajot, Martial Michel, and John S. Garofolo. The rich transcription 2006 spring meeting recognition evaluation. In *Machine Learning for Multimodal Interaction, Third International Workshop, MLMI 2006*, volume 4299 of *Lecture Notes in Computer Science*, pages 309–322, 2006.

[37] A Nagrani, JS Chung, and AP Zisserman. VoxCeleb: a large-scale speaker identification dataset. In *18th Annual Conference of the International Speech Communication Association, Interspeech 2017*, 2017.

[38] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny. Speaker adaptation of neural network acoustic models using i-vectors. In *IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2013*, pages 55–59, 2013.

[39] Miquel Àngel India Massana, José Adrián Rodríguez Fonollosa, and Francisco Javier Hernando Pericás. LSTM neural network-based speaker segmentation using acoustic and language modelling. In *18th Annual Conference of the International Speech Communication Association, Interspeech 2017*, pages 2834–2838, 2017.

[40] Marek Hrúz and Marie Kunešová. Convolutional neural network in the task of speaker change detection. In *International Conference on Speech and Computer, SPECOM 2016*, pages 191–198, 2016.

[41] Marek Hrúz and Zbyněk Zajíc. Convolutional neural network for speaker change detection in telephone speaker diarization system. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017*, pages 4945–4949, 2017.

[42] Lukas Mateju, Petr Cerva, and Jindrich Zdánskỳ. An approach to online speaker change point detection using DNNs and WFSTs. In *20th Annual Conference of the International Speech Communication Association, Interspeech 2019*, pages 649–653, 2019.

[43] Mickael Rouvier, Grégor Dupuy, Paul Gay, Elie Khoury, Teva Merlin, and Sylvain Meignier. An open-source state-of-the-art toolbox for broadcast news diarization. In *14th Annual Conference of the International Speech Communication Association, Interspeech 2013*, pages 1477–1481, 2013.

[44] Sylvain Meignier, Jean-François Bonastre, and Stéphane Igounet. E-HMM approach for learning and adapting sound models for speaker indexing. In *The Speaker and Language Recognition Workshop (Odyssey 2001)*, pages 175–180, 2001.

[45] Amit S Malegaonkar, Aladdin M Ariyaeeinia, and Perasiriyan Sivakumaran. Efficient speaker change detection using adapted Gaussian mixture models. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(6):1859–1869, 2007.

[46] Fabio Castaldo, Daniele Colibro, Emanuele Dalmasso, Pietro Laface, and Claudio Vair. Stream-based speaker segmentation using speaker factors and eigenvoices. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2008*, pages 4133–4136, 2008.

[47] Vishwa Gupta. Speaker change point detection using deep neural nets. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015*, pages 4420–4424, 2015.

[48] Leda Sarı, Samuel Thomas, Mark Hasegawa-Johnson, and Michael Picheny. Pre-training of speaker embeddings for low-latency speaker change detection in broadcast news. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019*, pages 6286–6290, 2019.

[49] Neil Zeghidour, Olivier Teboul, and David Grangier. Dive: End-to-end speech diarization via iterative speaker embedding. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 702–709, 2021.

[50] Awni Hannun. Sequence modeling with CTC. *Distill*, 2(11), 2017.

[51] 1996 English Broadcast News Speech (HUB4). https://catalog.ldc.upenn.edu/LDC97S44.

[52] 1996 English Broadcast News Transcripts (HUB4). https://catalog.ldc.upenn.edu/LDC97T22.

[53] J. S. Chung, A. Nagrani, and A. Zisserman. VoxCeleb2: Deep speaker recognition. In *19th Annual Conference of the International Speech Communication Association, Interspeech 2018*, pages 1086–1090, 2018.

[54] Tanel Alumäe. The TalTech system for the VoxCeleb Speaker Recognition Challenge 2020. Technical report, Tallinn University of Technology, 2020.

[55] David Snyder, Guoguo Chen, and Daniel Povey. MUSAN: A music, speech, and noise corpus. *arXiv preprint arXiv:1510.08484*, 2015.

[56] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur. A study on data augmentation of reverberant speech for robust speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017*, pages 5220–5224, 2017.

[57] I. Szöke, M. Skácel, L. Mošner, J. Paliesek, and J. Černocký. Building and evaluation of a real room impulse response dataset. *IEEE Journal of Selected Topics in Signal Processing*, 13(4):863–876, 2019.

[58] Yue Fan, JW Kang, LT Li, KC Li, HL Chen, ST Cheng, PY Zhang, ZY Zhou, YQ Cai, and Dong Wang. CN-CELEB: a challenging Chinese speaker recognition dataset. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020*, pages 7604–7608, 2020.

[59] Daniel Garcia-Romero and Carol Y Espy-Wilson. Analysis of i-vector length normalization in speaker recognition systems. In *12th Annual Conference of*

the International Speech Communication Association, Interspeech 2011, pages 249–252, 2011.

[60] Hervé Bredin and Antoine Laurent. End-to-end speaker segmentation for overlap-aware resegmentation. In *22nd Annual Conference of the International Speech Communication Association, Interspeech 2021*, pages 3111–3115, 2021.

[61] Neville Ryant, Prachi Singh, Venkat Krishnamohan, Rajat Varma, Kenneth Church, Christopher Cieri, Jun Du, Sriram Ganapathy, and Mark Liberman. The Third DIHARD Diarization Challenge. In *22nd Annual Conference of the International Speech Communication Association, Interspeech 2021*, pages 3570–3574, 2021.

[62] Juan Manuel Coria, Hervé Bredin, Sahar Ghannay, and Sophie Rosset. Overlap-aware low-latency online speaker diarization based on end-to-end local segmentation. In *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2021*, pages 1139–1146, 2021.

[63] Tanel Alumäe, Joonas Kalda, Külliki Bode, and Martin Kaitsa. Automatic closed captioning for Estonian live broadcasts. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 492–499, 2023.

[64] Zhuo Chen, Takuya Yoshioka, Liang Lu, Tianyan Zhou, Zhong Meng, Yi Luo, Jian Wu, Xiong Xiao, and Jinyu Li. Continuous speech separation: Dataset and analysis. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020*, pages 7284–7288, 2020.

[65] Thilo von Neumann, Keisuke Kinoshita, Marc Delcroix, Shoko Araki, Tomohiro Nakatani, and Reinhold Haeb-Umbach. All-neural online source separation, counting, and diarization for meeting analysis. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019*, pages 91–95, 2019.

[66] Soumi Maiti, Yushi Ueda, Shinji Watanabe, Chunlei Zhang, Meng Yu, Shi-Xiong Zhang, and Yong Xu. EEND-SS: joint end-to-end neural speaker diarization and speech separation for flexible number of speakers. In *IEEE Spoken Language Technology Workshop, SLT 2022*, pages 480–487, 2022.

[67] Alon Vinnikov, Amir Ivry, Aviv Hurvitz, Igor Abramovski, Sharon Koubi, Ilya Gurvich, Shai Peer, Xiong Xiao, Benjamin Martinez Elizalde, Naoyuki Kanda, Xiaofei Wang, Shalev Shaer, Stav Yagev, Yossi Asher, Sunit Sivasankaran, Yifan Gong, Min Tang, Huaming Wang, and Eyal Krupka. NOTSOFAR-1 challenge: New datasets, baseline, and tasks for distant meeting transcription. In *25th Annual Conference of the International Speech Communication Association, Interspeech 2024*, pages 5003–5007, 2024.

[68] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022.

[69] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R. Hershey. SDR - half-baked or well done? In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019*, pages 626–630, 2019.

[70] Hervé Bredin. pyannote.audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe. In *24th Annual Conference of the International Speech Communication Association, Interspeech 2023*, pages 1983–1987, 2023.

[71] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François Grondin, William Aris, Hwidong Na, Yan Gao, Renato De Mori, and Yoshua Bengio. SpeechBrain: A general-purpose speech toolkit. *arXiv preprint arXiv:2106.04624*, 2021.

[72] Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. w2v-BERT: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2021*, pages 244–250, 2021.

[73] Seamless Communication, Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady ElSahar, Justin Haaheim, John Hoffman, Min-Jae Hwang, Hirofumi Inaguma, Christopher Klaiber, Ilia Kulikov, Pengwei Li, Daniel Licht, Jean Maillard, Ruslan Mavlyutov, Alice Rakotoarison, Kaushik Ram Sadagopan, Abinesh Ramakrishnan, Tuan Tran, Guillaume Wenzek, Yilin Yang, Ethan Ye, Ivan Evtimov, Pierre Fernandez, Cynthia Gao, Prangthip Hansanti, Elahe Kalbassi, Amanda Kallet, Artyom Kozhevnikov, Gabriel Mejia Gonzalez, Robin San Roman, Christophe Touret, Corinne Wong, Carleigh Wood, Bokai Yu, Pierre Andrews, Can Balioglu, Peng-Jen Chen, Marta Ruiz Costa-jussà, Maha Elbayad, Hongyu Gong, Francisco Guzmán, Kevin Heffernan, Somya Jain, Justine T. Kao, Ann Lee, Xutai Ma, Alexandre Mourachko, Benjamin Peloquin, Juan Pino, Sravya Popuri, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, Anna Y. Sun, Paden Tomasello, Changhan Wang, Jeff Wang, Skyler Wang, and Mary Williamson. Seamless: Multilingual expressive and streaming speech translation. *arXiv preprint arXiv:2312.05187*, 2023.

[74] Shengkui Zhao, Yukun Ma, Chongjia Ni, Chong Zhang, Hao Wang, Trung Hieu Nguyen, Kun Zhou, Jia Qi Yip, Dianwen Ng, and Bin Ma. MossFormer2: Combining transformer and RNN-Free recurrent network for enhanced time-domain monaural speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2024*, pages 10356–10360, 2024.

[75] Shiliang Zhang, Cong Liu, Hui Jiang, Si Wei, Lirong Dai, and Yu Hu. Feed-forward sequential memory networks: A new structure to learn long-term dependency. *arXiv preprint arXiv:1512.08301*, 2015.

[76] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Maël Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried M. Post, Dennis Reidsma, and Pierre Wellner. The AMI meeting corpus: A pre-announcement. In *Machine Learning for Multimodal Interaction, Second International Workshop, MLMI 2005*, volume 3869 of *Lecture Notes in Computer Science*, pages 28–39, 2005.

[77] Fan Yu, Shiliang Zhang, Yihui Fu, Lei Xie, Siqi Zheng, Zhihao Du, Weilong Huang, Pengcheng Guo, Zhijie Yan, Bin Ma, Xin Xu, and Hui Bu. M2Met: The Icassp 2022 multi-channel multi-party meeting transcription challenge. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022*, pages 6167–6171, 2022.

[78] Shinji Watanabe, Michael Mandel, Jon Barker, Emmanuel Vincent, Ashish Arora, Xuankai Chang, Sanjeev Khudanpur, Vimal Manohar, Daniel Povey, Desh Raj, David Snyder, Aswin Shanmugam Subramanian, Jan Trmal, Bar Ben Yair, Christoph Boeddeker, Zhaoheng Ni, Yusuke Fujita, Shota Horiguchi, Naoyuki Kanda, Takuya Yoshioka, and Neville Ryant. CHiME-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings. In *6th International Workshop on Speech Processing in Everyday Environments (CHiME 2020)*, pages 1–7, 2020.

[79] Thilo von Neumann, Christoph B. Boeddeker, Marc Delcroix, and Reinhold Haeb-Umbach. MeetEval: A toolkit for computation of word error rates for meeting transcription systems. In *7th International Workshop on Speech Processing in Everyday Environments (CHiME 2023)*, pages 27–32, 2023.

[80] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[81] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[82] Jiangyu Han, Federico Landini, Johan Rohdin, Anna Silnova, Mireia Diez, and Lukáš Burget. Leveraging self-supervised learning for speaker diarization. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2025*, pages 1–5, 2025.

[83] Marc Härkönen, Samuel J. Broughton, and Lahiru Samarakoon. EEND-M2F: Masked-attention mask transformers for speaker diarization. In *25th Annual Conference of the International Speech Communication Association, Interspeech 2024*, pages 37–41, 2024.

[84] Naoyuki Kanda, Jian Wu, Yu Wu, Xiong Xiao, Zhong Meng, Xiaofei Wang, Yashesh Gaur, Zhuo Chen, Jinyu Li, and Takuya Yoshioka. Streaming multi-talker ASR with token-level serialized output training. In *23rd Annual Conference of the International Speech Communication Association, Interspeech 2022*, pages 3774–3778, 2022.

[85] Edresson Casanova, Julian Weber, Christopher D Shulby, Arnaldo Candido Junior, Eren Gölge, and Moacir A Ponti. YourTTS: Towards zero-shot multi-speaker TTS and zero-shot voice conversion for everyone. In *International Conference on Machine Learning, ICML 2022*, pages 2709–2720, 2022.

[86] Naohiro Tawara, Marc Delcroix, Atsushi Ando, and Atsunori Ogawa. NTT speaker diarization system for CHiME-7: Multi-domain, multi-microphone end-to-end and vector clustering diarization. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2024*, pages 11281–11285, 2024.

[87] Séverin Baroudi, Thomas Pellegrini, and Hervé Bredin. Specializing self-supervised speech representations for speaker segmentation. In *25th Annual Conference of the International Speech Communication Association, Interspeech 2024*, pages 3769–3773, 2024.

[88] Salah Zaiem, Youcef Kemiche, Titouan Parcollet, Slim Essid, and Mirco Ravanelli. Speech self-supervised representations benchmarking: a case for larger probing heads. *Computer Speech & Language*, 89:101695, 2025.

[89] Alexis Plaquet and Hervé Bredin. Powerset multi-class cross entropy loss for neural speaker diarization. In *24th Annual Conference of the International Speech Communication Association, Interspeech 2023*, pages 3222–3226, 2023.

[90] Yihui Fu, Luyao Cheng, Shubo Lv, Yukai Jv, Yuxiang Kong, Zhuo Chen, Yanxin Hu, Lei Xie, Jian Wu, Hui Bu, Xin Xu, Jun Du, and Jingdong Chen. AISHELL-4: An open source dataset for speech enhancement, separation, recognition and speaker diarization in conference scenario. In *22nd Annual Conference of the International Speech Communication Association, Interspeech 2021*, pages 1736–1740, 2021.

[91] Tao Liu, Shuai Fan, Xu Xiang, Hongbo Song, Shaoxiong Lin, Jiaqi Sun, Tianyuan Han, Siyuan Chen, Binwei Yao, Sen Liu, Yifei Wu, Yanmin Qian, and Kai Yu. MSDWild: Multi-modal speaker diarization dataset in the wild. In *23rd Annual Conference of the International Speech Communication Association, Interspeech 2022*, pages 1476–1480, 2022.

[92] Zehui Yang, Yifan Chen, Lei Luo, Runyan Yang, Lingxuan Ye, Gaofeng Cheng, Ji Xu, Yaohui Jin, Qingqing Zhang, Pengyuan Zhang, Lei Xie, and Yonghong Yan. Open source MagicData-RAMC: A rich annotated Mandarin conversational (RAMC) speech dataset. In *23rd Annual Conference of the International Speech Communication Association, Interspeech 2022*, pages 1736–1740, 2022.

[93] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, M. Martin, T. Nagarajan, I. Radosavovic, S. K. Ramakrishnan, F. Ryan, J. Sharma, M. Wray, M. Xu, E. Z. Xu, C. Zhao, S. Bansal, D. Batra, V. Cartillier, S. Crane, T. Do, M. Doulaty, A. Erapalli, C. Feichtenhofer, A. Fragomeni, Q. Fu, C. Fuegen, A. Gebreselasie, C. Gonzalez, J. Hillis, X. Huang, Y. Huang, W. Jia, W. Khoo, J. Kolar, S. Kottur, A. Kumar, F. Landini, C. Li, Y. Li, Z. Li, K. Mangalam, R. Modhugu, J. Munro, T. Murrell, T. Nishiyasu, W. Price, P. R. Puentes, M. Ramazanova, L. Sari, K. Somasundaram, A. Southerland, Y. Sugano, R. Tao, M. Vo, Y. Wang, X. Wu, T. Yagi, Y. Zhu, P. Arbelaez, D. Crandall, D. Damen, G. M. Farinella, B. Ghanem, V. K. Ithapu, C. V. Jawahar, H. Joo, K. Kitani, H. Li, R. Newcombe, A. Oliva, H. S. Park, J. M. Rehg, Y. Sato, J. Shi, M. Z. Shou, A. Torralba, L. Torresani, M. Yan, and J. Malik. Ego4D: Around the World in 3,000 Hours of Egocentric Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*, pages 18995—-19012, 2022.

[94] Arsha Nagrani, Joon Son Chung, Jaesung Huh, Andrew Brown, Ernesto Coto, Weidi Xie, Mitchell McLaren, Douglas A Reynolds, and Andrew Zisserman. VoxSRC 2020: The second VoxCeleb speaker recognition challenge. *arXiv preprint arXiv:2012.06867*, 2020.

[95] Hervé Bredin, Ruiqing Yin, Juan Manuel Coria, Gregory Gelly, Pavel Korshunov, Marvin Lavechin, Diego Fustes, Hadrien Titeux, Wassim Bouaziz, and Marie-Philippe Gill. Pyannote.audio: Neural building blocks for speaker diarization. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020*, pages 7124–7128, 2020.

[96] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 2623–2631, 2019.

[97] Federico Landini, Mireia Diez, Alicia Lozano-Diez, and Lukáš Burget. Multi-speaker and wide-band simulated conversations as training data for end-to-end neural diarization. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2023*, pages 1–5, 2023.

[98] Samuele Cornell, Jordan Darefsky, Zhiyao Duan, and Shinji Watanabe. Generating data with text-to-speech and large-language models for conversational speech recognition. *arXiv preprint arXiv:2408.09215*, 2024.

Acknowledgements

Completing this thesis would not have been possible without the support and guidance of many individuals. I am sincerely thankful to everyone who contributed to my academic growth and personal well-being during this endeavor.

I am grateful to my supervisor, Tanel, for introducing me to the fascinating field of speech processing and for his steady support, availability, and encouragement of independent thinking. I also appreciate the TalTech NLP group for providing an engaging and supportive environment in which to learn and exchange ideas.

I thank Hervé for his guidance during the second half of my PhD. The collaboration that began during my visit to Toulouse greatly influenced the direction of this work, and I thank the IRIT team for their warm welcome and stimulating research environment.

I am also thankful to Ricard for his expertise and guidance in navigating the domain of speech separation.

Finally, I am grateful to my family and friends for their support throughout this process, and especially to Aishah for her constant belief in me.

Abstract

The proliferation of multi-speaker audio content—from conference recordings and broadcast media to conversational AI systems—has created an urgent need for robust automatic processing of complex acoustic scenes. While single-speaker speech processing has achieved remarkable success, the transition to multi-speaker environments introduces fundamental challenges that current methodologies struggle to address effectively.

This thesis investigates core problems in multi-talker speech processing (MTSP), with a particular focus on the domain mismatch between training conditions and real-world deployment scenarios. We propose novel methodologies that bridge the gap between controlled single-speaker settings and the acoustic complexity of natural conversations, where speakers overlap, interrupt each other, and exhibit diverse acoustic characteristics.

The primary contributions of this work span three interconnected areas:

- collar-aware training methodologies that align model optimization with evaluation protocols,
- joint training frameworks that leverage real-world multi-speaker recordings for both diarization and separation tasks,
- robust speaker embedding techniques that maintain discriminability and improve processing speed in overlapping speech scenarios.

These contributions address fundamental limitations in current MTSP systems and demonstrate substantial improvements in realistic evaluation conditions.

Kokkuvõte

Mitme rääkijaga helimaterjali tekib aina juurde, alates koosolekusalvestistest kuni vestluslike tehisintellektisüsteemideni. See on omakorda suurendanud vajadust keerukate akustiliste stseenide usaldusväärseks automaattöötluseks. Kui üksikkõneleja kõne töötlus on saavutanud märkimisväärseid tulemusi, siis üleminek mitme rääkijaga keskkondadesse toob kaasa põhimõttelisi väljakutseid, millega senised meetodid ei suuda tõhusalt toime tulla.

Käesolev väitekiri uurib mitme rääkijaga kõne töötluse (MRKT) keskseid probleeme, keskendudes eriti domeenierinevusele treeningutingimuste ja reaalse kasutuskeskkonna vahel. Pakutakse välja uusi meetodeid, mis aitavad ületada lõhet kontrollitud üksikkõneleja korpuste ja loomulike vestluste vahel, kus kõnelejad räägivad üksteisega samaaegselt, katkestavad üksteist ning kus esineb suur akustiline varieeruvus.

Antud väitekiri panustab kolme omavahel seotud uurimissuunda:

- kraetundlikud treeningumeetodid, mis viivad mudeli treenimise vastavusse hindamisprotokollidega;
- ühistreeningraamistikud, mis kasutavad mitme rääkijaga salvestisi nii diariseerimise kui ka eraldamise ülesanneteks;
- töökindlad kõnelejaesituste meetodid, mis säilitavad eristusvõime ja suurendavad töötlemiskiirust kattuva kõne korral.

Need panused aitavad ületada MRKT-süsteemide seniseid kitsaskohti ning tõstavad täpsust realistlikes hindamistingimustes.

Appendix 1

I

Joonas Kalda and Tanel Alumäe. Collar-aware training for streaming speaker change detection in broadcast speech. In *Odyssey 2022: The Speaker and Language Recognition Workshop*, pages 141–147, 2022

Collar-aware Training for Streaming Speaker Change Detection in Broadcast Speech

Joonas Kalda, Tanel Alumäe

Department of Software Science
Tallinn University of Technology, Estonia
joonas.kalda@taltech.ee, tanel.alumae@taltech.ee

Abstract

In this paper, we present a novel training method for speaker change detection models. Speaker change detection is often viewed as a binary sequence labelling problem. The main challenges with this approach are the vagueness of annotated change points caused by the silences between speaker turns and imbalanced data due to the majority of frames not including a speaker change. Conventional training methods tackle these by artificially increasing the proportion of positive labels in the training data. Instead, the proposed method uses an objective function which encourages the model to predict a single positive label within a specified collar. This is done by marginalizing over all possible subsequences that have exactly one positive label within the collar. Experiments on English and Estonian datasets show large improvements over the conventional training method. Additionally, the model outputs have peaks concentrated to a single frame, removing the need for post-processing to find the exact predicted change point which is particularly useful for streaming applications.

1. Introduction

Speaker change detection (SCD) is a task of locating precise points in the audio recording when a different speaker starts speaking. It is often used as the first step in speaker diarization systems. Depending on the application, SCD systems can be either streaming (also known as *online*) or batch-processing (*offline*). In a batch processing system, the whole audio recording is available when SCD is applied. This allows the model to use all information from both past and future frames when locating speaker change points. A streaming model, on the other hand, needs to identify speaker change points with low latency, using typically only one or two seconds of audio from the future. Streaming SCD is needed as a preprocessing step in streaming speech recognition systems that perform unsupervised speaker adaptation, e.g. using i-vectors [1], so that the speaker adaptation state could be reset at speaker change points. SCD is also often an explicit requirement in realtime closed captioning systems for broadcast television [2].

Most modern SCD systems are based on supervised learning. Large speech datasets, manually annotated with speaker change points, are used for training and SCD is treated as a binary sequence classification task. Long short-term memory (LSTM) recurrent neural networks [3, 4] or convolutional neural networks [5, 6, 7] are often used as models. An important issue when training such models for SCD is that the annotated change points in the training data are ambiguous and imbalanced. The ambiguousness comes from the fact that often there is a substantial amount of silence between the speech of two adjacent speakers, yet only a single frame is marked as a change point. The choice where exactly the annotated change point resides is often inconsistent, resulting in training data that is confusing for the model. Also, the number of frames in the training data labelled as change points is usually less than 1% of all the frames, causing problems with model convergence.

In this paper, we propose a novel objective function for training sequence classification models for SCD. This *collar-aware* objective function gives the SCD model more freedom by allowing it to choose an appropriate speaker change point within the neighbourhood of the annotated change point. This method addresses both the problems of imbalanced data as well as the ambiguousness of the annotated labels. Furthermore, the models trained using this method are especially well suited for streaming applications, as the resulting model generates "peaky" change points that do not require any post-processing to find local maxima. We show that the method also achieves notably higher accuracy in both streaming and batch-processing scenarios, compared to several well-established baselines[1].

2. Related work

SCD approaches can be divided into two main categories: metric- and model based. The first approach operates by applying a pair of sliding windows on the sequence of feature vectors extracted from the underlying audio signal and uses a divergence metric for comparing their contents. A speaker change point is detected if the divergence between two adjacent windows is larger than a predefined threshold and the divergence achieves a significant local maximum. The advantage of this method is that it doesn't require a large annotated training corpus for training: only the value of the threshold parameter needs finetuning on a small validation set. This method is used in many speaker diarization systems that use Gaussian mixture models (GMMs) as their main building blocks (e.g. [8])

A model based approach, on the other hand, uses a a training corpus with manually annotated speaker change points to train a model for this task. Many different models have been proposed, such as hidden Markov models [9], GMMs [10], eigenvoices [11], deep neural networks (DNNs) [12, 7], convolutional neural networks [5, 6, 7], recurrent neural networks [3, 4] and Siamese networks [13].

Since models based on neural networks have become popular in recent years for this task, we review three approaches based on them more carefully. In [4], SCD is formulated as a standard binary sequence labelling task that can be tackled using bidirectional LSTMs: the model's task is to label each

[1] Code and demo available at https://github.com/alumae/online_speaker_change_detector

frame with either 0 (no speaker change) or 1 (speaker change). One problem with this approach is that the training data is heavily imbalanced: the number of frames that are labelled with 0 is much larger than the number of frames labelled with 1 (only 0.4% according to [4]). Under standard training, the model converges to a state in which a 0 is predicted for each frame. Therefore, [4] increases the number of positive labels artificially by labelling frames 50 ms on each side of the annotated change point as 1. During inference, local score maxima exceeding a pre-determined threshold are marked as speaker change points.

In [7], a somewhat similar approach is used, but instead of a bidirectional LSTM, a CNN is used that "sees" a fixed-size window of feature frames prior and after the current frame. This allows operating the model with low latency in streaming mode. As with the LSTM-based approach, a large number of frames in the direct neighbourhood of the annotated change point are labelled as positive during model training, in order to make the training data more balanced.

In [13], a Siamese architecture is used for low-latency SCD: a 2-second window prior and after the current frame is processed by a bidirectional LSTM, resulting in two embedding vectors. The embeddings are then processed by a classification module that decides whether the two segments correspond to different speakers. Various pretraining schemes can be applied to the embedding computation module that are found to improve the detection performance by a large amount. This work handles the imbalanced data problem by sampling a predefined ratio of speaker change points from the training data to each batch.

Inconsistent and unreliable speaker turn boundaries in manually annotated training data can also have a negative effect on the performance of end-to-end speaker diarization systems. In [14], a modification to the standard multilabel classification loss for speaker diarization is introduced that simply ignores the errors in a defined radius around annotated speaker change points.

3. Collar-aware training

Speaker change detection is often regarded as a binary sequence labelling problem. We consider an audio recording consisting of feature vectors x_i for $i = 1, ..., N$ and the corresponding speaker boundary labels $y_i \in \{0, 1\}$ with $y_i = 1$ meaning that the frame corresponds to an annotated speaker boundary.

When a SCD system is evaluated in terms of precision and recall of detected speaker boundaries, it is a standard practice to use a *collar* (typically 250 ms) for annotated speaker boundaries: if the boundary detected by the model is within the tolerated amount of milliseconds of the annotated boundary, the detected speaker change point is assumed to be correct. However, under standard sequence labelling objective (Figure 1, *a*), the collar is not used, making the training objective different from the evaluation scenario.

As pointed out in the previous section, several papers [4, 7] have suggested to artificially modify the training data of the speaker boundary detection model by labelling a predefined number of frames around the annotated speaker boundary as additional (pseudo-)boundaries (Figure 1, *b*). This is done in order to make the training data more balanced in terms of label frequencies, and to model the inherent ambiguousness of the speaker boundaries.

We propose to use a modified objective function for training SCD models that solves both the problems of imbalanced data and ambiguous annotated boundaries. Instead of labelling points around the annotated boundary as pseudo-boundaries, it supervises the model to label exactly one frame within the given collar as a speaker boundary, but the exact position of the boundary can be freely chosen (Figure 1, *c*). This method has several advantages: (1) it matches the evaluation criteria better than method (*b*); (2) it solves the imbalanced data problem similarly or better than method (*b*); (3) the model trained in this manner can be easily applied in online mode without any post-processing to find the local maximum, since the output of the model is now very "peaky" (see Section 4.5.1).

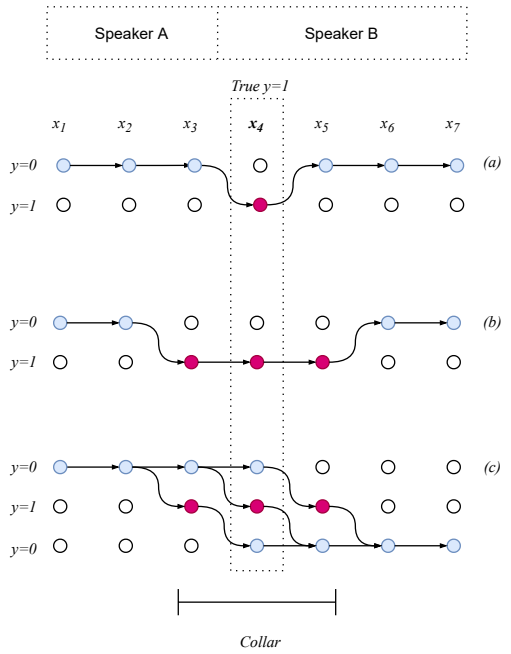

Figure 1: **Outline of three supervision methods for speaker change detection:** *(a)* corresponds to standard sequence labelling objective; *(b)* increases the number of positive labels artificially by setting several frames in the neighbourhood of the annotated change point as positive [4]; *(c)* the proposed method sums over all paths that have exactly one positive label in the neighbourhood of the annotated change point.

Formally, given the reference labels y and model predictions $\hat{y}_i$, the standard binary sequence labelling objective is:

$$\mathcal{L}(\hat{y}, y) = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)$$

Since the boundaries occur very sparsely, this objective can be efficiently calculated by summing over the log likelihoods of the no-boundary events, and then modifying it to account for the few boundary events. Given annotated boundary positions $Z = \{z | y_z = 1\}$, the standard sequence labelling loss becomes:

$$\mathcal{L}(\hat{y}, Z) = -\Big(\sum_{i=1}^{N} \log(1 - \hat{y}_i) - \sum_{z_i \in Z} \log(1 - \hat{y}_{z_i}) + \sum_{z_i \in Z} \log(\hat{y}_{z_i}) \Big)$$

```
1  def collar_bce_loss(log_probs, change_points, collar):
2    """
3    Compute collar-aware binary CE loss.
4
5    Arguments:
6    log_probs -- tensor of shape (seq_len, 2), containing log likelihoods of non-boundary
7      and boundary events
8    change_points -- indexes of annotated boundaries
9    collar -- value of the collar (in frames)
10   """
11   result = log_probs[:, 0].sum()
12   for change_point in change_points:
13     collar_variant_logs = []
14     collar_start_i = change_point - collar
15     collar_end_i = change_point + collar
16     time_index = range(collar_start_i, collar_end_i + 1)
17     event_index = torch.eye(collar_end_i - collar_start_i + 1).long()
18     collar_variant_logprobs = log_probs[time_index, event_index].sum(1)
19     result -= log_probs[time_index, 0].sum()
20     result += torch.logsumexp(collar_variant_logprobs, 0)
21   return -result
```

Figure 2: Pytorch code for efficient calculation of the collar-aware binary cross-entropy loss.

In order to calculate the proposed collar-aware objective, we have to consider a superset $S(Z)$ of all sets of boundary events Z' where for each original change point $z_i \in Z$ there is exactly one boundary event that is within it's collar set $C_i = \{x | z_i - c < x < z_i + c\}$, where c is the value of the collar. Alternatively,

$$S(Z) = \{\{z_1, ..., z_N\} | z_i \in C_i \forall i \in \{1, ..., N\}\},$$

where N is the total number of original change points. For example, if the reference label sequence is [00100], then Z' at collar $= 2$ corresponds to a set of label sequences $\{[01000], [00100], [00010]\}$.

The proposed objective sums over all such change point configurations:

$$\mathcal{L}_{\text{collar}}(\hat{y}, Z) = -\log \sum_{Z' \in S(Z)} \mathrm{e}^{-\mathcal{L}(\hat{y}, Z')}$$

The idea of this objective function is somewhat similar to the CTC loss function [15] used for training end-to-end speech recognition models. As with CTC, it is not practical to compute it using brute force. In order to make it more efficient, we can again use our knowledge that the number of speaker boundaries occur very sparsely. Figure 2 lists the Pytorch implementation of this idea. The collar-aware loss can be calculated by summing over the log-likelihoods of the non-boundary events (line 11), subtracting the log likelihoods of the non-boundary events that lie within a boundary collar (line 19), and then adding the marginalized log-likelihood of having exactly one boundary somewhere within the collar (line 20).

4. Experiments

4.1. Datasets

The experiments were carried on both English and Estonian datasets. For English, we used the HUB4 speech dataset [16, 17]. The Estonian dataset consists of TV and radio broadcasts.

Table 1: A comparison of lengths and the number of speaker change points in the datasets used in the experiments.

Dataset	Train	Development	Test
Estonian	497.2h / 80k	1.2h / 166	0.7h / 102
English	128.3h / 19.5k	6.1h / 893	5.4h / 893

Both datasets are manually transcribed and annotated with speaker information. Test and development data were separated similarly for both datasets: 10 recordings were chosen for each at random. An overview of dataset sizes and annotated boundary counts is provided in Table 1. The datasets are similarly balanced, with 0.04% of the frames being labelled as speaker change points.

4.2. Implementation details

We consider two different architectures which we train using both the standard training method and the proposed collar-aware one.

The first architecture was chosen to closely resemble that of [4]. 33-dimensional acoustic features are extracted every 10ms on a 25ms window, consisting of 11-dimensional MFCCs and their first and second derivatives. The model is made up of two Bi-LSTM layers having 64 outputs and 40 outputs and a multi-layer-perceptron with 40-, 10- and 1-dimensional layers.

The second architecture uses a Resnet-based feature extractor before a LSTM layer. The Resnet module is extracted from a speaker recognition model pretrained on VoxCeleb2 [18], as described in [19]. It results in 1280-dimensional features with a frame subsampling rate of 8. In the low-latency streaming model, the Resnet layer is followed by two 256-dimensional LSTM layers, and 1-second label delay is used in order for the model to see the data past the current frame (see Figure 3). In the offline model, the LSTMs are replaced with bidirectional LSTMs, and no label delay is used.

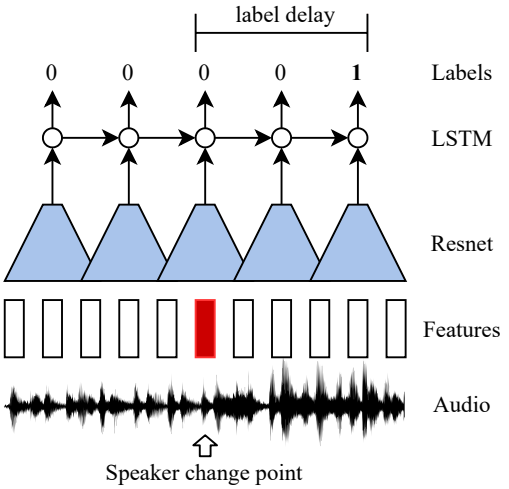

Figure 3: **Architecture of the streaming *Resnet-LSTM* model:** Filterbank features from speech frames are fed into a Resnet module that is pretrained on a speaker recognition task and then trained jointly with the rest of the model. The outputs from the Resnet module are fed into a LSTM layer that identifies speaker boundaries ($\texttt{label} = 1$). Since this cannot be done without the knowledge of future frames, the identification is done with a label delay that corresponds to 1 second of speech (in streaming mode).

All our training methods use extracted segments with random lengths between 10s and 30s.

The first training method used also follows [4]. Namely the training data is artificially modified by positively labelling every frame in a 50ms neighborhood of an annotated change point. Notably, no additional labelling is needed for the Resnet-based architecture since the subsampling that happens during feature extraction results in frames of 80ms duration and thus a 50ms neighborhood corresponds to roughly a single frame. A standard binary sequence labelling objective is used as the loss function for this method.

The second training method includes no artificial labelling and instead uses the proposed collar-aware objective as the loss function. The size of the collar was chosen to be $c = 250$ms and the effects of varying the collar size are discussed in Section 4.5.2.

During training, data augmentation is applied: background noise and/or reverberation is added to each training segment, both with a probability of 0.3. The background noises originate from the MUSAN corpus [20]. For reverberation, we used simulated small and medium room impulse responses [21] and real room impulse responses from the BUT Speech@FIT Reverb Database [22].

4.3. Evaluation metrics

The evaluation metrics are standard precision and recall calculated on the test sets. Predicted change points are considered correct if they match an annotated change point within a forgiveness collar (closest pairs are matched first until no pairs remain). Although our main evaluation metrics are precision and recall of detected speaker change points at a forgiveness collar of 250 ms, we also show the same metrics using a larger 0.5 second collar. A change point is predicted to happen if the local maximum of the models output is higher than a threshold, the value of which is determined by maximizing the F1 score on the development set.

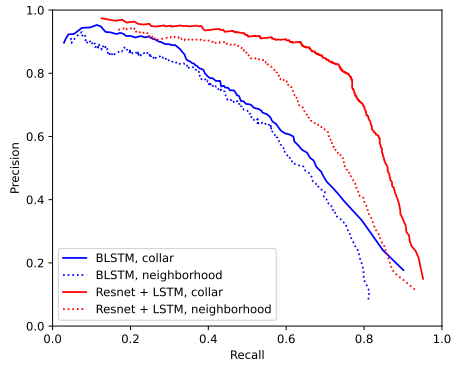

Figure 4: Precision-recall curves for models trained on the English dataset.

4.4. Baselines

In addition to the pure LSTM-based speaker segmentation system [4], we compare our results to various baselines. Since speaker change points can be easily derived from the output of a speaker diarization system, we use several speaker diarization models that have achieved competitive results on various diarization benchmarks.

The recently proposed VBx diarization method [23] has produced state-of-the-art results on CALLHOME, AMI and DIHARD II datasets. The method uses a Bayesian hidden Markov model to find speaker clusters in a sequence of x-vectors. We used the open source implementation of the method available at GitHub[2]. The diarization pipeline first extracts x-vectors from the sections of the audio that contain speech. The provided x-vector models are trained on VoxCeleb1 [24], VoxCeleb2 [18] and CN-CELEB [25]. The x-vectors are extracted every 0.25 seconds from overlapping sub-segments of 1.5 seconds. The x-vectors are centered, whitened and length normalized [26]. The x-vectors are pre-clustered using agglomerative hierarchical clustering to obtain the initial speaker labels and finally further clustered using the VBx model. The used the VBx parameters F_A, F_b and P_{loop} tuned on the respective development sets in order to minimize the boundary detection F1 score with a 250 ms forgiveness collar.

We also compare to the neural speaker segmentation method implemented in *pyannote.audio* [27] that performs joint voice activity detection, speaker segmentation and overlapped speech detection. Similarly to the original EEND approach [28], here speaker segmentation is modeled as a multi-label classification problem using permutation-invariant training. The model op-

[2] `https://github.com/BUTSpeechFIT/VBx`

Table 2: Precision (P), recall (R) and F1 results of various batch-mode and streaming models on Estonian and English datasets with two two different forgiveness collar values.

	Estonian dataset						English dataset					
	collar=0.25s			collar=0.50s			collar=0.25s			collar=0.50s		
Model	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Batch-mode processing												
Pretrained speaker diarization (VBx)	0.68	0.68	0.68	0.96	0.96	**0.96**	0.48	0.64	0.55	0.67	0.88	0.76
Pretrained *pyannote.audio*	0.62	0.73	0.67	0.68	0.79	0.73	0.42	0.38	0.40	0.57	0.51	0.54
+ finetuned on the given dataset	0.82	0.82	0.82	0.89	0.89	0.89	0.60	0.49	0.54	0.73	0.59	0.65
BLSTM	0.7	0.85	0.77	0.74	0.88	0.81	0.44	0.59	0.50	0.50	0.62	0.55
+ collar aware training	0.75	0.81	0.78	0.86	0.80	0.83	0.59	0.57	0.58	0.61	0.61	0.61
Resnet + BLSTM	0.80	0.78	0.79	0.84	0.80	0.82	0.59	0.66	0.62	0.65	0.69	0.67
+ collar aware training	0.92	0.89	**0.91**	0.96	0.92	0.94	0.76	0.69	**0.73**	0.79	0.76	**0.78**
Streaming processing												
pyannote.audio with latency=1.0s	0.34	0.67	0.45	0.37	0.73	0.49	0.21	0.33	0.26	0.28	0.44	0.34
+ finetuned on our data	0.42	0.68	0.51	0.46	0.75	0.57	0.26	0.45	0.32	0.30	0.52	0.38
Resnet + LSTM	0.73	0.73	0.73	0.76	0.75	0.76	0.56	0.62	0.59	0.58	0.71	0.64
+ collar aware training	0.89	0.83	**0.86**	0.92	0.86	**0.89**	0.66	0.71	**0.68**	0.72	0.75	**0.74**

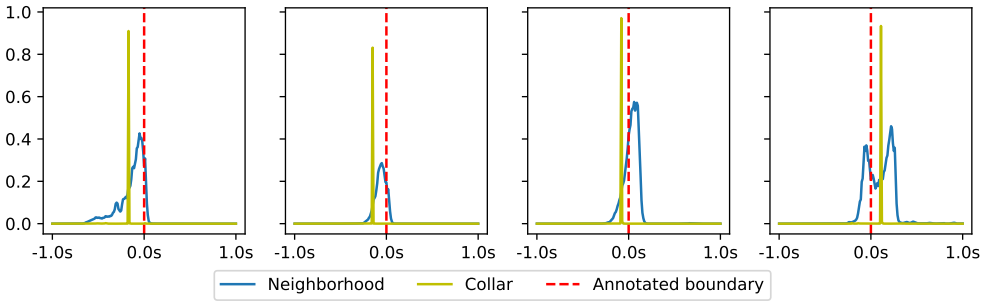

Figure 5: Random samples of Resnet+BLSTM model outputs for neighborhood- and collar-based models trained on the English dataset centered around annotated speaker change points.

erates on short audio chunks (5 seconds) at a temporal resolution of every 16 ms and outputs speaker activation probabilities that are stitched together across frames. More specifically, we use the model available at `https://huggingface.co/pyannote/segmentation` that is trained on the DIHARD3 corpus [29, 30]. We also experiment with the same model in a low-latency setting [31], using the open-source implementation[3]. In streaming mode, the latency of the segmentation output is configurable. To make the results comparable to our streaming model, we used a latency of 1 second.

In addition to using the publicly available *pyannote.audio* segmentation model, we also experimented with finetuning it on our training data. This was done on each dataset and resulted in further baselines for both streaming and offline settings.

In order to convert the output of the diarization systems to speaker change points we consider all the consecutive pairs of speaker segments where the speaker ids differ. If there is less than 2 seconds between the two segments then a speaker change point is predicted at the beginning of the second segment. This was found to give better results than other choices in the gap like the midpoint or the end of the first segment.

4.5. Results

A comparison of the model performances on the two datasets is provided in Table 2. The results are divided into two categories: models that perform change point detection in batch mode, and streaming models. The Resnet+LSTM based models trained using the proposed collar-aware loss function clearly outperform the same models trained using the standard training method on both datasets. Furthermore, these models also provide higher speaker change point detection accuracy than the baseline speaker diarization models. The state-of-the-art VBx diarization model actually results in impressive accuracy at a collar of 0.5 seconds but much lower accuracy at the standard

[3] `https://github.com/juanmc2005/StreamingSpeakerDiarization/`

0.25 second collar. This might be due to the fact that the VBx model uses a relatively large temporal resolution of 0.25 seconds which causes the detected change point to be considered an error if it is off by just one timestep.

Precision-recall curves obtained by varying the classification threshold on the English test set are presented in Figure 4. It can be seen that collar-aware training outperforms neighbourhood-based training at all operation points for both LSTM and Resnet-BLSTM based models.

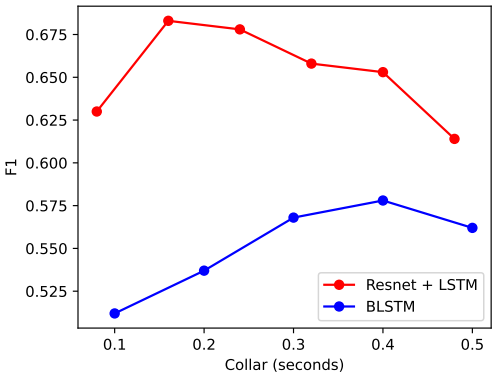

Figure 6: F1 scores on the English dataset for models of varying training collar size.

4.5.1. *Peakiness of model output*

One benefit of the collar-aware loss-function discussed above was the "peaky" output of the model. Figure 5 demonstrates this effect by visualizing samples obtained from Resnet+BLSTM model outputs centered around randomly chosen annotated boundaries for the English dataset. The output obtained from a model trained using the neighborhood-based method is spread out over multiple frames requiring finding the exact local maximum in post-processing. In comparison, the change points predicted by the model trained with the collar-based method can be obtained by simply comparing the model outputs to a threshold since the activations tend to be limited to a single frame.

4.5.2. *Tuning the collar size*

Figure 6 shows the influence of the collar size used during training on the F1 score on English test data. The optimal size for the collar is dependant on the nature of the data, how imbalanced it is and how reliable the annotated boundaries are. Overall, there seems to be flexibility to the choice of collar size as the F1 score does not change a lot across the tested range. Notably, all of the tested collar sizes lead to a better result than the neighborhood based models.

5. Conclusion

This work presented a novel supervision method for speaker change detection models using a collar-aware objective function. In our experiments we compared it with a conventional training method that artificially labels a neighborhood of an annotated boundary as positive as well as various state-of-the-art speaker diarization models. We find that our collar-aware training yields improved results both for a purely LSTM-based model and one that uses pretrained embeddings with 8-fold subsampling.

We analyzed model outputs around randomly chosen boundaries to show that the activations for our method are concentrated to a single frame. This makes our training method well suited for online applications as there is no need for local maxima detection in post-processing.

The exact choice of collar size was determined to not have a great effect on performance with choices from 80ms to 500ms all outperforming the conventional training method.

6. Acknowledgements

The authors acknowledge the TalTech supercomputing resources made available for conducting the research reported in this paper.

7. References

[1] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *ASRU*, 2013.

[2] Hagai Aronowitz and Weizhong Zhu, "Context and uncertainty modeling for online speaker change detection," in *ICASSP*, 2020.

[3] Miquel Àngel India Massana, José Adrián Rodríguez Fonollosa, and Francisco Javier Hernando Pericás, "LSTM neural network-based speaker segmentation using acoustic and language modelling," in *Interspeech*, 2017.

[4] Ruiqing Yin, Hervé Bredin, and Claude Barras, "Speaker change detection in broadcast TV using bidirectional long short-term memory networks," in *Interspeech*, 2017.

[5] Marek Hrúz and Marie Kunešová, "Convolutional neural network in the task of speaker change detection," in *International Conference on Speech and Computer*, 2016.

[6] Marek Hrúz and Zbyněk Zajíc, "Convolutional neural network for speaker change detection in telephone speaker diarization system," in *ICASSP*, 2017.

[7] Lukas Mateju, Petr Cerva, and Jindrich Zdánskỳ, "An approach to online speaker change point detection using DNNs and WFSTs," in *Interspeech*, 2019.

[8] Mickael Rouvier, Grégor Dupuy, Paul Gay, Elie Khoury, Teva Merlin, and Sylvain Meignier, "An open-source state-of-the-art toolbox for broadcast news diarization," in *Interspeech*, 2013.

[9] Sylvain Meignier, Jean-François Bonastre, and Stéphane Igounet, "E-HMM approach for learning and adapting sound models for speaker indexing," in *2001: A Speaker Odyssey-The Speaker Recognition Workshop*, 2001.

[10] Amit S Malegaonkar, Aladdin M Ariyaeeinia, and Perasiriyan Sivakumaran, "Efficient speaker change detection using adapted Gaussian mixture models," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 6, pp. 1859–1869, 2007.

[11] Fabio Castaldo, Daniele Colibro, Emanuele Dalmasso, Pietro Laface, and Claudio Vair, "Stream-based speaker segmentation using speaker factors and eigenvoices," in *ICASSP*, 2008.

[12] Vishwa Gupta, "Speaker change point detection using deep neural nets," in *ICASSP*, 2015.

[13] Leda Sarı, Samuel Thomas, Mark Hasegawa-Johnson, and Michael Picheny, "Pre-training of speaker embeddings for low-latency speaker change detection in broadcast news," in *ICASSP*, 2019.

[14] Neil Zeghidour, Olivier Teboul, and David Grangier, "DIVE: End-to-end speech diarization via iterative speaker embedding," *arXiv:2105.13802*, 2021.

[15] Awni Hannun, "Sequence modeling with CTC," *Distill*, vol. 2, no. 11, 2017.

[16] "1996 English Broadcast News Speech (HUB4)," https://catalog.ldc.upenn.edu/LDC97S44.

[17] "1996 English Broadcast News Transcripts (HUB4)," https://catalog.ldc.upenn.edu/LDC97T22.

[18] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Interspeech*, 2018.

[19] Tanel Alumäe, "The TalTech system for the VoxCeleb Speaker Recognition Challenge 2020," Tech. Rep., 2020.

[20] David Snyder, Guoguo Chen, and Daniel Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, arXiv:1510.08484v1.

[21] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *ICASSP*, 2017, pp. 5220–5224.

[22] I. Szöke, M. Skácel, L. Mošner, J. Paliesek, and J. Černocký, "Building and evaluation of a real room impulse response dataset," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 863–876, 2019.

[23] Federico Landini, Ján Profant, Mireia Diez, and Lukáš Burget, "Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: theory, implementation and analysis on standard tasks," *Computer Speech & Language*, vol. 71, pp. 101254, 2022.

[24] A Nagrani, JS Chung, and AP Zisserman, "VoxCeleb: a large-scale speaker identification dataset," in *Interspeech*, 2017.

[25] Yue Fan, JW Kang, LT Li, KC Li, HL Chen, ST Cheng, PY Zhang, ZY Zhou, YQ Cai, and Dong Wang, "CN-CELEB: a challenging Chinese speaker recognition dataset," in *ICASSP*. IEEE, 2020, pp. 7604–7608.

[26] Daniel Garcia-Romero and Carol Y Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Interspeech*, 2011.

[27] Hervé Bredin and Antoine Laurent, "End-to-end speaker segmentation for overlap-aware resegmentation," in *Interspeech*, 2021.

[28] Yusuke Fujita, Naoyuki Kanda, Shota Horiguchi, Kenji Nagamatsu, and Shinji Watanabe, "End-to-end neural speaker diarization with permutation-free objectives," in *Interspeech*, 2019.

[29] Neville Ryant, Prachi Singh, Venkat Krishnamohan, Rajat Varma, Kenneth Church, Christopher Cieri, Jun Du, Sriram Ganapathy, and Mark Liberman, "The third DIHARD diarization challenge," *arXiv preprint arXiv:2012.01477*, 2020.

[30] Neville Ryant, Kenneth Church, Christopher Cieri, Jun Du, Sriram Ganapathy, and Mark Liberman, "Third DIHARD challenge evaluation plan," *arXiv preprint arXiv:2006.05815*, 2020.

[31] Juan Manuel Coria, Hervé Bredin, Sahar Ghannay, and Sophie Rosset, "Overlap-aware low-latency online speaker diarization based on end-to-end local segmentation," in *IEEE Automatic Speech Recognition and Understanding Workshop*, 2021.

Appendix 2

II

Joonas Kalda, Clément Pagés, Ricard Marxer, Tanel Alumäe, and Hervé Bredin. PixIT: Joint training of speaker diarization and speech separation from real-world multi-speaker recordings. In *Odyssey 2024: The Speaker and Language Recognition Workshop*, pages 115–122, 2024

PixIT: Joint Training of Speaker Diarization and Speech Separation from Real-world Multi-speaker Recordings

Joonas Kalda[1], *Clément Pagés*[2], *Ricard Marxer*[3], *Tanel Alumäe*[1], *Hervé Bredin*[2]

[1]Tallinn University of Technology, Estonia
[2]IRIT, Université de Toulouse, CNRS, Toulouse, France
[3]Université de Toulon, Aix Marseille Univ, CNRS, LIS, Toulon, France

Abstract

A major drawback of supervised speech separation (SSep) systems is their reliance on synthetic data, leading to poor real-world generalization. Mixture invariant training (MixIT) was proposed as an unsupervised alternative that uses real recordings, yet struggles with over-separation and adapting to long-form audio. We introduce PixIT, a joint approach that combines permutation invariant training (PIT) for speaker diarization (SD) and MixIT for SSep. With a small extra requirement of needing SD labels during training, it solves the problem of over-separation and allows stitching local separated sources leveraging existing work on clustering-based neural SD. We measure the quality of the separated sources via applying automatic speech recognition (ASR) systems to them. PixIT boosts the performance of various ASR systems across two meeting corpora both in terms of the speaker-attributed and utterance-based word error rates while not requiring any fine-tuning.

1. Introduction

Speech separation is the task of estimating individual speaker sources from a mixture. It is an important part of automatic speech technologies for meeting recordings as a significant proportion of the speech can be overlapped. Supervised training approaches, mainly permutation invariant training (PIT), have been shown to perform well on few seconds long fully-overlapped synthetic speech mixtures that fit in the memory for the model [1, 2]. To extend a PIT-based approach to more realistic data, [3] proposed the task of continuous speech separation (CSS). This involves generating long-form separated sources from a continuous audio stream that contains multiple utterances that partially overlap. The standard method for extending PIT-based separation systems to CSS is by applying them on a sliding window and reordering sources in neighboring chunks based on a similarity metric calculated on the overlapped region. In long-form audio, however, the speaker tracking breaks down if a speaker stops speaking for longer than the overlapping portion of the sliding window.

Another problem of PIT-based training that remains in CSS approaches is the reliance on clean single-speaker isolated sources for the synthetic mixtures. The supervised approach does not generalize well to real-world data as clean ground truth separated reference signals are not available in recordings due to cross-talk. To combat this, mixture invariant training (MixIT) was introduced in [4], an unsupervised method that does not require clean separated sources for training. Two mixtures from the target domain are added together to obtain a mixture of mixtures (MoM) and a separation model is trained to estimate sources so that they can be combined to obtain the original mixtures. In [5] it was demonstrated that this method is effective in using real-world meetings as the target domain. A limitation of MixIT is that the number of output sources for the separation model has to be twice the maximum number of speakers of a single mixture. This can lead to over-separation and makes it difficult to generalize to long-form audio. Over-separation can be mitigated by performing semi-supervised training but this still relies on synthetic data. In [6], MixIT was used in combination with speaker diarization pre-processing to perform source separation on real-world long-form meeting audio. Separation was done at the utterance level and the correct speaker sources to use were determined by comparing speaker embeddings with global embeddings obtained from diarization. This resulted in superior speaker-attributed automatic speech recognition (ASR) performance. A limitation of this approach is the need for extra voice activity detection (VAD) and speaker diarization models to segment long-form audio into speaker-attributed utterances, as speech separation is performed solely at the utterance level.

Traditional speaker diarization approaches have relied on a multi-step approach consisting of VAD to obtain speaker segments, local speaker embeddings, and clustering [7]. End-to-end diarization (EEND) is a newer approach that is able to handle overlapped speech but comes with its own limitations, such as needing a large amount of data and mispredicting the number of speakers [8, 9]. Recently the two approaches have been combined into the *best-of-both-worlds* framework [10, 11] which performs EEND on small chunks and stitches the results together using speaker embeddings and clustering.

Speech separation and speaker diarization are both often parts of multi-speaker automatic transcription systems. The models used to carry out these two tasks are mostly cascaded in two different ways. Since the sources extracted by a speaker separation system no longer have speech overlap regions, they can greatly facilitate the speaker diarization task improving its performance. An example of such a system is the speaker separation guided diarization system (SSGD) [12, 13]. A drawback of this method is that diarization depends on the quality of the separated sources. Another option is to place a diarization system upstream of a speaker separation system, like in [14, 15]. Indeed, source separation is easier if the speech activity of each speaker is known, provided that the diarization system is able to manage speech overlap. Similarly to the previous approach, the speech separation performance depends on the quality of the speaker diarization. Thus, we can see that these two tasks can benefit from the results of the other, highlighting their interdependence, and the fact that there is no obvious choice whether to start the processing with a diarization or speech separation system. This has served as motivation

for joint learning approaches. The Recurrent Selective Attention Network architecture (RSAN) [16] was the first all-neural model to jointly perform the speech separation, speaker diarization, and speaker counting tasks. In this model, the extraction is made over time using sliding blocks. In each block, speakers are iteratively extracted from the mixture by estimating a mask for each of them, given speaker embeddings determined in the previous blocks, and a residual mask from the previous iterations in the current block. Another architecture that performs jointly these three tasks is the end-to-end neural diarization and speech separation architecture (EEND-SS) [17]. This system is based on the EEND framework for the diarization and speaker counting tasks and Conv-TasNet [1] for the speaker separation one. In the EEND-SS architecture, the information given by the diarization branch is used to refine the separation part, by providing an estimation of the number of speakers and using the probability of speech activity to enhance the separated source signals. These joint approaches, however, still all rely on synthetic data for separation training.

We propose a joint framework for performing both speaker diarization and speech separation on long-form real-world audio. We name the approach PixIT, as it combines PIT for speaker diarization and MixIT for speech separation. We leverage speaker diarization information that is often available for meeting corpora to create MoMs that have the maximum number of speakers limited to better mimic real-world mixtures. Our separation/diarization model processes the mixture/MoM and outputs separated source predictions and the respective speaker activity predictions. When training the joint model we combine the PIT-loss for both the original mixtures and MoMs with the MixIT loss for the MoM. Aligning speaker sources with the speaker activations also solves the over-separation problem of MixIT. In inference, we are able to stitch together the separated sources across the sliding windows by first stitching the speaker activations as is done in the *best-of-both-worlds* approach for diarization. To measure the quality of the long-form stitched separated sources, we feed them into a variety of off-the-shelf ASR systems. We observe improvements over the baseline method of speaker attribution done through diarization for all ASR systems and two real-word meeting datasets: AMI [18] and AliMeeting [19]. Furthermore, we show that when the speaker-attributed transcripts are combined into a single output, the utterance-wise word error rate (uWER) improves.

2. Joint model

We base our model on the TasNet architecture [20], which consists of a 1-D convolutional encoder, a separator module that predicts N masking matrices and a 1-D convolutional decoder. We additionally leverage pre-trained WavLM features [21] which are especially suited for speech separation due to the use of the utterance mixing augmentation in their pre-training. These are concatenated with the convolutional encoder outputs. The diarization network takes the encoded separated signals as input and processes each source independently effectively performing VAD. The independent processing of the sources in the diarization module is required to maintain alignment between the separation outputs and the diarization branches. The joint model architecture, which we call ToTaToNet[1], is illustrated in Figure 1. The components of the model related to the **diarization** branch are colored **orange**, the components related to **separation** are colored **purple** and the components used by **both branches** are colored a **gradient** between the two. This color scheme is kept consistent across all the figures in the paper.

[1]Collaboration between labs in **To**ulouse, **Ta**llinn, and **To**ulon

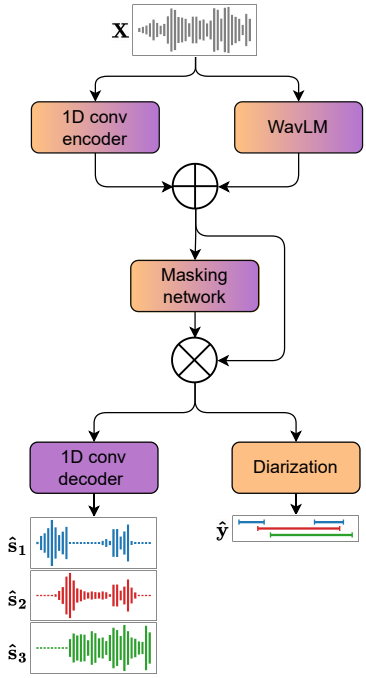

Figure 1: The architecture of the proposed ToTaToNet model.

2.1. Training

The joint training method for speech separation and speaker diarization is illustrated in Figure 2. Consider an audio chunk $\mathbf{X}$ and the reference speaker activity labels $\mathbf{y} \in \{0,1\}^{K_{\max} \times T}$ where $y_{k,t} = 1$ if speaker k is active at frame t and $y_{k,t} = 0$ otherwise. Here $K_{\max}$ specifies the maximum number of speakers anticipated in an audio chunk. For diarization, we utilize the well-established permutation-invariant training (PIT) objective [8]:

$$\mathcal{L}_{\text{PIT}}(\mathbf{y}, \hat{\mathbf{y}}) = \min_{\mathbf{P}} \sum_{k=1}^{K_{\max}} \mathcal{L}_{\text{BCE}}\left(\mathbf{y_k}, [\mathbf{P}\hat{\mathbf{y}}]_k\right),$$

where $\hat{\mathbf{y}}$ are the predicted speaker activations and $\mathbf{P}$ is an $K_{\max} \times K_{\max}$ permutation matrix and $\mathcal{L}_{\text{BCE}}$ is the standard binary cross entropy loss.

Using the speaker annotations, we construct two audio chunks $(\mathbf{X^1}, \mathbf{y^1})$ and $(\mathbf{X^2}, \mathbf{y^2})$ with non-overlapping sets of speakers with the total number of speakers no greater than $K_{\max}$. Limiting the total number of speakers is critical in solving the over-separation issue of MixIT. The MoM is constructed as $\mathbf{X}^{\text{MoM}} = \mathbf{X^1} + \mathbf{X^2}$ and the corresponding speaker activity labels $\mathbf{y}^{\text{MoM}}$ are given by $y^{\text{MoM}}_{:,t} = (y^1_{:,t}, y^2_{:,t})$ where the rows corresponding to non-active speakers are removed so that $\mathbf{y}^{\text{MoM}} \in \{0,1\}^{K_{\max} \times T}$. Then the MixIT loss function is given by,

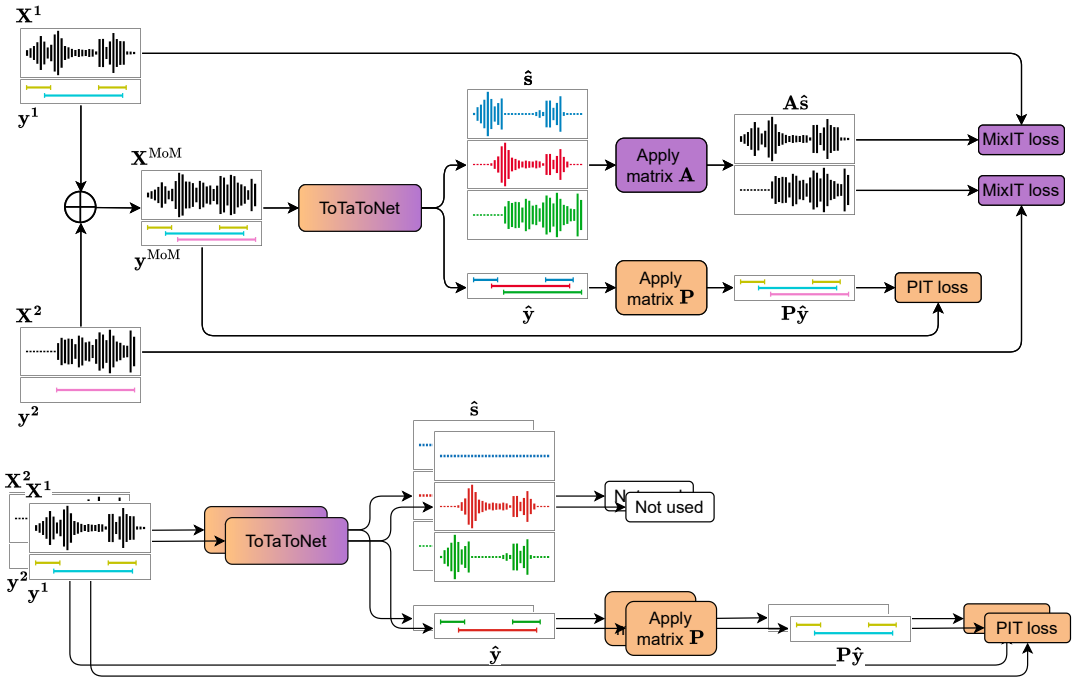

Figure 2: Training the joint model. The upper part shows calculating the MixIT and PIT losses on MoMs. The bottom part shows calculating PIT losses on the original mixtures.

$$\mathcal{L}_{\text{MixIT}}\left(\{\mathbf{X_n}\},\hat{\mathbf{s}}\right)=\min_{\mathbf{A}}\sum_{n=1}^{2}\mathcal{L}_{\text{SI-SDR}}\left(\mathbf{X_n},[\mathbf{A}\hat{\mathbf{s}}]_n\right),$$

where $\hat{\mathbf{s}}$ are the predicted separated sources, M is the number of output sources and $\mathbf{A}$ is a mixing matrix $\mathbf{A}\in\{0,1\}^{2\times M}$ under the constraint that each column sums to 1 and $\mathcal{L}_{\text{SI-SDR}}$ is the negative scale-invariant signal-to-distortion ratio [22]. Thanks to how we limit the total number of speakers when sampling the mixtures, we are able to use a significantly lower value for M.

Our combined multi-task loss is,

$$\mathcal{L}_{\text{PixIT}}=\lambda\Big(\mathcal{L}_{\text{PIT}}(\mathbf{y^1},\hat{\mathbf{y}}^{\mathbf{1}})+\mathcal{L}_{\text{PIT}}(\mathbf{y^2},\hat{\mathbf{y}}^{\mathbf{2}})$$
$$+\mathcal{L}_{\text{PIT}}(\mathbf{y}^{\text{MoM}},\hat{\mathbf{y}}^{\text{MoM}})\Big)+(1-\lambda)\mathcal{L}_{\text{MixIT}}\left(\{\mathbf{X_n}\},\hat{\mathbf{s}}\right),$$

where among the three values, 0.1, 0.5, and 0.9, $\lambda=0.5$ was selected due to its superior performance on the development data.

2.2. Inference

During inference, an audio stream is partitioned into shorter chunks as depicted in Figure 3. The joint model processes each chunk and outputs aligned estimates for speaker sources and speaker activations. The resulting speaker activations and corresponding sources are clustered as in [23]. First, speaker activations are binarized using a detection threshold $\theta\in[0,1]$ to obtain speaker segments. Second, local speaker embeddings are extracted from each chunk for all the active speakers. We only utilize the regions of the chunk where the corresponding speaker is active. Speaker embeddings are computed by feeding the concatenation of original audio samples corresponding to those regions to the pre-trained ECAPA-TDNN model [24] available in [25]. Finally, agglomerative hierarchical clustering is performed on these embeddings using a clustering threshold δ. As an important post-processing step, we perform leakage removal by setting the stitched separated sources at time t to zero when the diarization outputs predict that the corresponding speaker is not active and has not been active in a window $[t-\Delta t,t+\Delta t]$. This is a key benefit of the aligned speaker activations and speaker sources since it eliminates all cross-talk when the corresponding speaker is not active. The goal of introducing Δt is to give downstream ASR systems additional context. The hyperparameters θ, δ, and Δt are optimized on the development dataset.

3. Experiments

3.1. Datasets

We chose two publicly-available real-world meeting datasets AMI and AliMeeting for our experiments. AMI [18] consists of roughly 100 hours of English data. AliMeeting [19] is a Mandarin Chinese dataset with approximately 120 hours of recordings. As our goal is single-channel speech separation we only use the first channel of the microphone array also known as the single distant microphone (SDM) audio from AMI and channel 1 from AliMeeting for our experiments. Table 1 shows statistics for the two datasets [15]. While both datasets consist of meeting recordings, AliMeeting contains significantly more overlap.

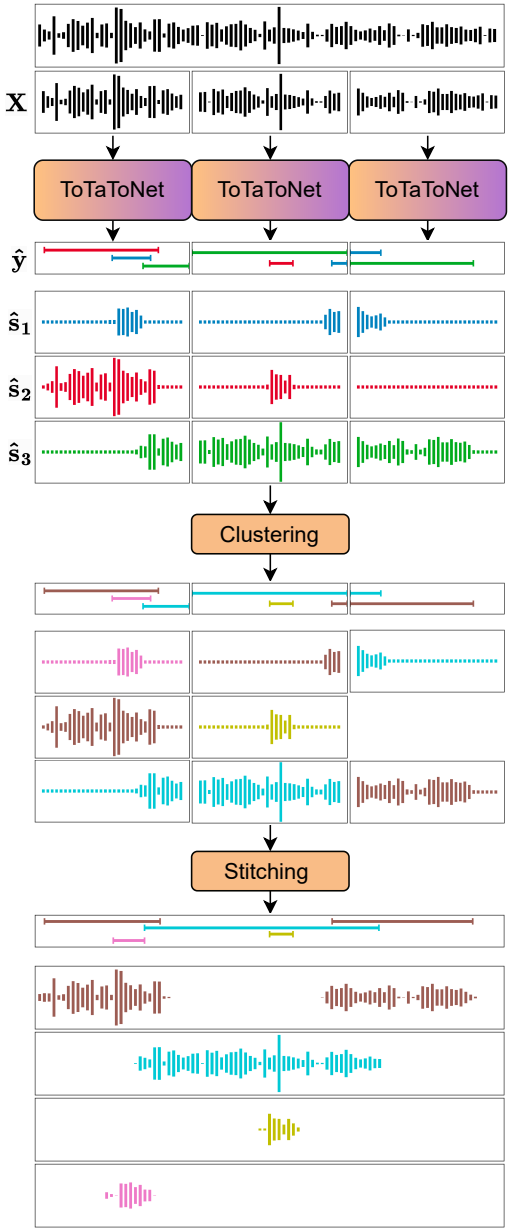

Figure 3: Inference on long-form audio. For ease of visualization inference using non-overlapping sliding windows is shown.

In all our experiments, the ToTaToNet model is trained only on the train set of the corresponding dataset.

3.2. Evaluation

Metrics. As ground-truth reference sources are not available for real-world data we use ASR performance as a proxy for evaluating the quality of long-form separation. We apply an ASR system independently on each of the separated sources and report the word error rate (WER) between the speaker-attributed predictions and references. Multiple definitions of WER have been proposed for ASR systems that process audio with multiple speakers and output multiple word sequences (MIMO) [26]. We choose concatenated minimum permutation WER (cpWER) [27] as our main metric because it is the only one that penalizes speaker confusion which is unwanted for long-form speaker sources. On Mandarin data, this metric corresponds to the concatenated minimum-permutation character error rate (cpCER). We use the same text normalizer as Whisper for both English and Mandarin.

It is important to note that the definition of cpWER that we used does not penalize redundant hypothesis speaker channels [26]. For transparency we include results using a definition of cpWER that includes this penalization using MeetEval's implementation [28]. These can be found in Appendix A.

On English data, we also report the utterance-wise WER (uWER) which ignores speaker attribution. The uWERs are calculated using Kaldi scripts [29] which in turn utilize asclite [30]. When using the long-form separated sources as input the single transcript is generated by first concatenating the ASR predictions from all the long-form sources and then sorting the words by start time.

Our metric for evaluating the diarization performance is the diarization error rate (DER) [31], which is defined as the sum of false alarm, missed detection, and speaker confusion rates. No forgiveness collar is used.

ASR systems. To verify the quality of the separated sources we experiment with multiple ASR systems. For English data, we chose the small.en, medium.en, and large-v2 Whisper models [32] and NVIDIA's stt_en_conformer_ctc_large available in the NeMo toolkit [33] on the basis that they were among the top performers on AMI as indicated by [34]. On Mandarin data, we only tested the aforementioned Whisper models with the English-only variants replaced with multilingual ones.

Speaker attribution. When evaluating cpWER (or cpCER for Mandarin AliMeeting), we compare two methods of adding speaker attribution (SA) to an ASR system. One through long-form separated sources and the other through speaker diarization. In the first case, the ASR systems are applied on the long-form separated sources immediately yielding speaker-attributed transcripts. In the latter, an ASR system is

Table 1: Statistics of datasets used for evaluations. The k-speaker durations are in terms of fraction of total speaking time.

	AMI			AliMeeting		
	Train	Dev	Test	Train	Eval	Test
Duration (h:m)	79:23	9:40	9:03	111:21	4:12	10:46
Num. sessions	133	18	16	209	8	20
Silence (%)	18.1	21.5	19.6	7.11	7.7	8.0
1-speaker (%)	75.5	74.3	73.0	52.5	62.1	63.4
2-speaker (%)	21.1	22.2	21.0	32.8	27.6	24.9
>2-speaker (%)	3.4	3.5	6.0	14.7	10.2	11.7

used on the original audio, and the predicted utterances are divided between speakers according to a speaker diarization system. Namely, each utterance is attributed to the speaker whose speaking segments have the most overlap with it. In the rare case that multiple speakers have fully overlapping speaking segments with the utterance, it is randomly attributed to one of them. In the following, we will refer to these two approaches as SA methods and refer to the system that was used to perform either diarization or separation as the SA system.

Word timestamps. For experiments with the Whisper family of models, we utilized WhisperX [35] which has implemented word-level time-stamps using forced alignment with a wav2vec2.0-based phoneme model [36]. The NeMo toolkit also provides word-level timestamps for the stt_en_conformer_ctc_large model.

Baselines. Our baseline systems perform speaker attribution through the *pyannote.audio* 3.1 speaker diarization pipeline [37].

3.3. Implementation details

During training, we sample the first mixture randomly across all the annotated regions from all the training files. Then we sample the second mixture from the same file while ensuring that it has no speakers in common with the first mixture and the total number of speakers is not greater than the number of output sources of the model. Sampling the other chunk from the same file has two benefits. First, it is important that the two mixtures come from the same recording conditions, otherwise the model might learn to exploit this difference as found in [5]. Second, this approach generalizes better because it does not require dataset-wise consistent speaker IDs.

Our system is implemented in the *pyannote.audio* toolkit [23] with the help of the *Asteroid* library [38]. We use 5-second sliding windows with a step size of 500ms as in [23] and in line with [5]. For both AMI and AliMeeting, there is a less than 1% chance that a 5-second window contains more than three active speakers [37]. Motivated by this statistic and aiming to mitigate over-separation, we set $K_{\max} = 3$. As a consequence of our sampling method for the mixtures, training data does not include windows with more than three speakers.

In ToTaToNet, the 1D conv encoder and decoder use a kernel size of 32, a stride of 16, and 64 filters. We concatenate the encoder output with WavLM-large pre-trained features which have a stride of 320 so the WavLM features are repeated 20 times. For the separator module we chose a DPRNN [2] with chunk size 100, hop size 50, and the rest of the hyperparameters kept the same as in the original work. The diarization module starts with an 8-fold average pooling layer to decrease the temporal resolution to that of [23]. We follow it with a simple diarization model consisting of a fully connected neural network with two 64-dimensional layers. We thus rely on the masking network to do the bulk of the work for speaker diarization. Importantly, due to the PIT training for diarization, the diarization module has to process each encoded masked source separately (as does the 1D conv decoder) otherwise the diarization outputs might be permuted with respect to the separated sources.

We use a learning rate of $1e^{-5}$ for the WavLM parameters and $3e^{-4}$ for the rest of the parameters. The learning rate is halved whenever the validation loss plateaus for 5 epochs. We use the Adam optimizer [39] with the gradients clipped to a L_2-norm of 5 and train all models for 100 epochs.

Table 2: The cpWER (%) on AMI-SDM for various ASR systems with speaker attribution (SA) done through diarization or the joint model

ASR model	SA method	SA system	cpWER(%)				Relative Change
			sub	del	ins	**total**	
Whisper small.en	Diarization	pyannote 3.1	8.7	27.2	3.7	**39.6**	
	Diarization	PixIT	8.5	27.3	2.1	**37.9**	-4.3%
	Separation	PixIT	6.7	25.8	1.4	**33.9**	-14.4%
Whisper medium.en	Diarization	pyannote 3.1	7.4	28.0	3.4	**38.8**	
	Diarization	PixIT	7.3	27.8	2.0	**37.1**	-4.4%
	Separation	PixIT	5.9	25.8	1.2	**32.8**	-15.4%
Whisper large-v2	Diarization	pyannote 3.1	7.1	29.3	1.8	**38.3**	
	Diarization	PixIT	6.9	26.6	2.1	**35.7**	-6.7%
	Separation	PixIT	5.6	24.7	1.3	**31.7**	-17.2%
NeMo conformer large	Diarization	pyannote 3.1	11.5	36.0	1.4	**48.9**	
	Diarization	PixIT	13.3	33.9	1.3	**48.5**	-0.8%
	Separation	PixIT	13.4	24.6	1.4	**39.4**	-19.4%

Table 3: The uWER (%) on AMI-SDM for various ASR systems using either the original audio or the separated sources as input

ASR model	Input to ASR	uWER(%)				Relative change
		sub	del	ins	**total**	
Whisper small.en	Original audio	6.7	29.6	1.4	**37.6**	
	Separated sources	6.9	27.9	1.5	**36.3**	-3.5%
Whisper medium.en	Original audio	5.8	30.0	1.3	**37.1**	
	Separated sources	6.0	27.7	1.4	**35.1**	-5.4%
Whisper large-v2	Original audio	5.2	28.9	1.3	**35.4**	
	Separated sources	5.5	26.9	1.4	**33.8**	-4.5%
NeMo conformer large	Original audio	10.7	36.7	1.8	**49.3**	
	Separated sources	12.6	26.4	2.6	**41.6**	-15.6%

When optimizing for the hyperparameters Δt, θ, and δ, we used either cpWER/cpCER or DER as the target metric depending on whether the pipeline was used for separation or diarization.

To ensure reproducibility, the code for both training and inference using PixIT will be available in the open-source *pyannote.audio* library. The recipes and separated source samples will be publicly available at `github.com/joonaskalda/PixIT`.

3.4. Results

The cpWERs for the various ASR systems on AMI-SDM test set are shown in Table 2. We can see that long-form separation via PixIT significantly improves the quality of speaker-attributed transcripts across the variety of ASR systems used. Notably, the ASR systems are applied on the separated sources off-the-shelf with no fine-tuning required.

We also report the uWER scores using either the original audio or the separated sources in Table 3. Across the ASR models, the bulk of the WER improvement comes from deletions. Having the original audio as input the ASR models may miss the quieter speakers utterances during overlap and utilizing separated sources helps recover those.

Table 4 shows the cpCERs for the AliMeeting channel 1 dataset. We can see that the improvement from utilizing separated sources is greater than 20% across the tested ASR systems. Notably, the relative improvements are greater than they were for the corresponding models on AMI data even though WavLM has been pre-trained on English data. This can be explained by the greater percentage of overlap present in AliMeeting as mentioned in section 3.1.

Table 4: The cpCER (%) on Alimeeting channel 1 for various ASR systems with speaker attribution (SA) done through diarization or the joint model

ASR system	SA	SA model	cpCER(%)				Relative Change
			sub	del	ins	**total**	
Whisper small	Diarization	pyannote 3.1	23.4	35.6	9.6	**68.6**	
	Diarization	PixIT	23.3	35.1	9.5	**67.9**	-1.0%
	Separation	PixIT	16.2	33.4	4.4	**54.0**	-21.3%
Whisper medium	Diarization	pyannote 3.1	18.5	37.9	9.5	**65.9**	
	Diarization	PixIT	18.8	37.2	8.9	**64.9**	-1.5%
	Separation	PixIT	11.8	34.2	4.2	**50.3**	-23.7%
Whisper large-v2	Diarization	pyannote 3.1	17.6	38.0	9.5	**65.1**	
	Diarization	PixIT	18.1	37.3	9.0	**64.4**	-1.1%
	Separation	PixIT	10.6	33.6	4.0	**48.3**	-25.8%

Table 5: The cpWER (%) on AMI-SDM for speaker-attribution (SA) done through PixIT speech separation with different configurations of WavLM and leakage removal. Whisper medium.en is used for ASR and pyannote 3.1 diarization as the baseline SA method.

SA method	WavLM	Leakage removal	cpWER(%)				Relative change
			sub	del	ins	**total**	
pyannote 3.1			7.4	28.0	3.4	**38.8**	
PixIT separation	✗	✗	19.2	15.3	15.6	**50.1**	+29.1%
	✗	✓	6.4	28.1	1.7	**36.2**	-6.7%
	✓	✗	9.3	21.0	3.8	**34.1**	-12.1%
	✓	✓	5.9	25.8	1.2	**32.8**	-15.5%

In Table 5, we show the effects of adding the WavLM features and performing leakage removal through the diarization output on our system performance when performing SA-ASR on AMI-SDM with Whisper medium.en. The system without WavLM features clearly has issues with leakage, with a lot of it passing through the VAD component of WhisperX. When using WavLM features the effect of our leakage removal is smaller but it still outperforms using only WhisperX. Notably, a decrease in substitution errors from applying leakage removal can be observed in both cases. A possible explanation is that since the leakage removal reduces the length of the predicted text, some words in the reference that previously corresponded to substitution errors now count as deletion errors. This is verified by the fact that the decrease in substitution errors is smaller than the increase in deletion errors. This effective method for leakage removal is a further benefit of ToTaToNet's aligned outputs. Leveraging the WavLM features significantly improves our system's performance which makes sense given the relatively small amount of data we have access to for training and the utterance mixing component of the pre-training. Still, even without using pre-trained features, we are able to improve on the baseline of WhisperX.

We also analyze PixIT's speaker diarization performance by measuring the DERs on AMI-SDM and AliMeeting for various training and hyperparameter optimization strategies as shown in Table 6. For the systems optimized for cpWER, we use $\Delta t = 0$, as that represents the real diarization capabilities. We have included the state-of-the-art (SOTA) systems as of February 2024. For AliMeeting this is the pyannote 3.1 system utilizing powerset training [37] and for AMI-SDM it is the end-to-end diarization model leveraging the Mask2Former architecture proposed in [40]. The DER scores are broken down into false alarm (FA), missed detection (MD), and speaker confusion (SC) rates. Optimizing for cpWER yields lower FA values. This means that a higher speaker activation threshold θ is used and only segments for which the diarization branch is confident are considered for ASR. Optimizing the $\lambda = 0.5$ system for DER improves on the SOTA for both AliMeeting and AMI-SDM. Training our system for only the easier task of speaker diarization i.e. with $\lambda = 1$, we achieve a further boost to performance on both datasets.

Table 6: DER (%) comparison with state-of-the-art systems on AMI-SDM and AliMeeting channel 1 for different training strategies and ways of optimizing the hyperparameters θ, δ, and Δt. For the latter, the underlying ToTaToNet is kept the same.

	DER(%)			
AMI-SDM systems	FA	MD	SC	**total**
Härkönen et al. [40]				**18.9**
PixIT, $\lambda = 0.5$, optimized for cpWER	1.3	17.9	6	**25.3**
PixIT, $\lambda = 0.5$, optimized for DER	3.9	8.2	5.6	**17.7**
PixIT, $\lambda = 1$	4.4	7.2	5.5	**17.1**
AliMeeting systems				
Plaquet et al. [37]	3.7	10.4	9.2	**23.3**
PixIT, $\lambda = 0.5$, optimized for cpWER	2.7	13.2	12.4	**28.3**
PixIT, $\lambda = 0.5$, optimized for DER	5.8	7.3	8.3	**21.4**
PixIT, $\lambda = 1$	4.7	6.5	8.3	**19.5**

4. Conclusion

In this paper, we proposed PixIT, a novel approach for performing multitask training for speaker diarization and speech separation. This method does not depend on clean single-speaker individual sources, only requiring single-channel recordings with speaker diarization labels which are usually a part of annotation. The local separated source and diarization predictions of the proposed ToTaToNet model are aligned allowing for long-form inference via the *best-of-both-worlds* approaches that have been developed for speaker diarization. A further benefit of the aligned sources is that we can perform effective leakage removal by zeroing out inactive speaker sources. We perform various experiments to demonstrate the quality of the long-form separated sources obtained from real-world meeting data by using them as input for various ASR systems. Indeed, the cpWERs show significant improvements over the baseline of performing speaker attribution using speaker diarization with the improvements increasing with the proportion of overlapped speech present. Furthermore, we observe a decrease in utterance-based WER when the ASR outputs from separated sources are combined into a single transcript. These results come from using the ASR systems on the separated sources off the shelf with no fine-tuning required. Finally, we show that PixIT achieves state-of-the-art speaker diarization performance on both the AMI-SDM and AliMeeting datasets.

5. Acknowledgements

We would like to acknowledge Dr Yoshiaki Bando, for pointing out the ambiguity in cpWER definitions regarding scenarios involving an unknown number of speakers. The research reported in this paper was supported by the Agence de l'Innovation Défense under the grant number 2022 65 0079. This work

was granted access to the HPC resources of GENCI-IDRIS under the allocations AD011014274, as well as the TalTech supercomputing resources.

6. References

[1] Yi Luo and Nima Mesgarani, "Conv-TasNet: Surpassing ideal time-frequency magnitude masking for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[2] Yi Luo, Zhuo Chen, and Takuya Yoshioka, "Dual-path RNN: Efficient long sequence modeling for time-domain single-channel speech separation," in *ICASSP*, 2020.

[3] Zhuo Chen, Takuya Yoshioka, Liang Lu, Tianyan Zhou, Zhong Meng, Yi Luo, Jian Wu, Xiong Xiao, and Jinyu Li, "Continuous speech separation: Dataset and analysis," in *ICASSP*, 2020.

[4] Scott Wisdom, Efthymios Tzinis, Hakan Erdogan, Ron Weiss, Kevin Wilson, and John Hershey, "Unsupervised sound separation using mixture invariant training," in *NeurIPS*, 2020.

[5] Aswin Sivaraman, Scott Wisdom, Hakan Erdogan, and John R. Hershey, "Adapting speech separation to real-world meetings using mixture invariant training," in *ICASSP*, 2022.

[6] Yuang Li, Xianrui Zheng, and Philip C. Woodland, "Self-supervised learning-based source separation for meeting data," in *ICASSP*, 2023.

[7] Federico Landini, Ján Profant, Mireia Diez, and Lukáš Burget, "Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: Theory, implementation and analysis on standard tasks," *Computer Speech and Language*, vol. 71, pp. 101254, 2022.

[8] Yusuke Fujita, Naoyuki Kanda, Shota Horiguchi, Kenji Nagamatsu, and Shinji Watanabe, "End-to-end neural speaker diarization with permutation-free objectives," in *Interspeech*, 2019.

[9] Yusuke Fujita, Naoyuki Kanda, Shota Horiguchi, Yawen Xue, Kenji Nagamatsu, and Shinji Watanabe, "End-to-end neural speaker diarization with self-attention," in *ASRU*, 2019.

[10] Keisuke Kinoshita, Marc Delcroix, and Naohiro Tawara, "Advances in integration of end-to-end neural and clustering-based diarization for real conversational speech," in *Interspeech*, 2021.

[11] Keisuke Kinoshita, Marc Delcroix, and Naohiro Tawara, "Integrating end-to-end neural and clustering-based diarization: Getting the best of both worlds," in *ICASSP*, 2021.

[12] Xin Fang, Zhen-Hua Ling, Lei Sun, Shu-Tong Niu, Jun Du, Cong Liu, and Zhi-Chao Sheng, "A deep analysis of speech separation guided diarization under realistic conditions," in *APSIPA ASC*, 2021.

[13] Giovanni Morrone, Samuele Cornell, Desh Raj, Luca Serafini, Enrico Zovato, Alessio Brutti, and Stefano Squartini, "Low-latency speech separation guided diarization for telephone conversations," in *SLT*, 2022.

[14] Christoph Boeddeker, Aswin Shanmugam Subramanian, Gordon Wichern, Reinhold Haeb-Umbach, and Jonathan Le Roux, "TS-SEP: Joint diarization and separation conditioned on estimated speaker embeddings," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 1185–1197, 2024.

[15] Desh Raj, Daniel Povey, and Sanjeev Khudanpur, "GPU-accelerated guided source separation for meeting transcription," in *Interspeech*, 2023.

[16] Thilo von Neumann, Keisuke Kinoshita, Marc Delcroix, Shoko Araki, Tomohiro Nakatani, and Reinhold Haeb-Umbach, "All-neural online source separation, counting, and diarization for meeting analysis," in *ICASSP*, 2019.

[17] Soumi Maiti, Yushi Ueda, Shinji Watanabe, Chunlei Zhang, Meng Yu, Shi-Xiong Zhang, and Yong Xu, "EEND-SS: Joint end-to-end neural speaker diarization and speech separation for flexible number of speakers," in *SLT*, 2022.

[18] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al., "The AMI meeting corpus: A pre-announcement," in *ICMI*, 2005.

[19] Fan Yu, Shiliang Zhang, Yihui Fu, Lei Xie, Siqi Zheng, Zhihao Du, Weilong Huang, Pengcheng Guo, Zhijie Yan, Bin Ma, Xin Xu, and Hui Bu, "M2Met: The ICASSP 2022 multi-channel multi-party meeting transcription challenge," in *ICASSP*, 2022.

[20] Yi Luo and Nima Mesgarani, "TasNet: Time-domain audio separation network for real-time, single-channel speech separation," in *ICASSP*, 2018.

[21] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[22] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R. Hershey, "SDR – half-baked or well done?," in *ICASSP*, 2019.

[23] Hervé Bredin, "pyannote.audio 2.1 speaker diarization pipeline: Principle, benchmark, and recipe," in *Interspeech*, 2023.

[24] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," in *Interspeech*, 2020.

[25] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, et al., "Speechbrain: A general-purpose speech toolkit," *arXiv preprint arXiv:2106.04624*, 2021.

[26] Thilo von Neumann, Christoph Boeddeker, Keisuke Kinoshita, Marc Delcroix, and Reinhold Haeb-Umbach, "On word error rate definitions and their efficient computation for multi-speaker speech recognition systems," in *ICASSP*, 2023.

[27] Shinji Watanabe, Michael Mandel, Jon Barker, Emmanuel Vincent, Ashish Arora, Xuankai Chang, Sanjeev Khudanpur, Vimal Manohar, Daniel Povey, Desh Raj, David Snyder, Aswin Shanmugam Subramanian, Jan Trmal, Bar Ben Yair, Christoph Boeddeker, Zhaoheng Ni, Yusuke Fujita, Shota Horiguchi, Naoyuki Kanda, Takuya Yoshioka, and Neville Ryant, "CHiME-6 Challenge: Tackling Multispeaker Speech Recognition for Unsegmented Recordings," in *CHiME 2020*, 2020.

[28] Thilo von Neumann, Christoph B. Boeddeker, Marc Delcroix, and Reinhold Haeb-Umbach, "Meeteval: A toolkit for computation of word error rates for meeting transcription systems," in *CHiME*, 2023.

[29] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The Kaldi speech recognition toolkit," in *ASRU*, 2011.

[30] Jonathan G Fiscus, Jerome Ajot, Nicolas Radde, Christophe Laprun, et al., "Multiple dimension Levenshtein edit distance calculations for evaluating automatic speech recognition systems during simultaneous speech.," in *LREC'06*, 2006.

[31] Jonathan G Fiscus, Jerome Ajot, Martial Michel, and John S Garofolo, "The rich transcription 2006 spring meeting recognition evaluation," *MLMI*, 2006.

[32] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever, "Robust speech recognition via large-scale weak supervision," in *ICML*, 2023.

[33] Oleksii Kuchaiev, Jason Li, Huyen Nguyen, Oleksii Hrinchuk, Ryan Leary, Boris Ginsburg, Samuel Kriman, Stanislav Beliaev, Vitaly Lavrukhin, Jack Cook, et al., "NeMo: a toolkit for building AI applications using neural modules," *arXiv preprint arXiv:1909.09577*, 2019.

[34] Vaibhav Srivastav, Somshubra Majumdar, Nithin Koluguri, Adel Moumen, Sanchit Gandhi, Hugging Face Team, Nvidia NeMo Team, and SpeechBrain Team, "Open automatic speech recognition leaderboard," `https://huggingface.co/spaces/huggingface.co/spaces/open-asr-leaderboard/leaderboard`, 2023.

[35] Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman, "WhisperX: Time-accurate speech transcription of long-form audio," *Interspeech*, 2023.

[36] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *NeurIPS*, 2020.

[37] Alexis Plaquet and Hervé Bredin, "Powerset multi-class cross entropy loss for neural speaker diarization," in *Interspeech*, 2023.

[38] Manuel Pariente, Samuele Cornell, Joris Cosentino, Sunit Sivasankaran, Efthymios Tzinis, Jens Heitkaemper, Michel Olvera, Fabian-Robert Stöter, Mathieu Hu, Juan M. Martín-Doñas, David Ditter, Ariel Frank, Antoine Deleforge, and Emmanuel Vincent, "Asteroid: the PyTorch-based audio source separation toolkit for researchers," in *Interspeech*, 2020.

[39] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *CoRR*, 2014.

[40] Marc Härkönen, Samuel J Broughton, and Lahiru Samarakoon, "Eend-m2f: Masked-attention mask transformers for speaker diarization," *arXiv preprint arXiv:2401.12600*, 2024.

[41] Xianrui Zheng, Chao Zhang, and Philip C Woodland, "Tandem multitask training of speaker diarisation and speech recognition for meeting transcription," *Interspeech*, 2022.

[42] Naoyuki Kanda, Yashesh Gaur, Xiaofei Wang, Zhong Meng, and Takuya Yoshioka, "Serialized Output Training for End-to-End Overlapped Speech Recognition," *Interspeech*, 2020.

A. On the evolution of cpWER definitions

A.1. Review of cpWER definitions

Here we provide a brief literature review on cpWER definitions as of early 2023, which served as the basis for our choice of a cpWER variant that does not penalize overestimation.

The original cpWER definition, as proposed in CHiME-6, was limited to scenarios involving a fixed number of speakers [27]. In our work, we adopted the extended cpWER definition from [26], which offered a comprehensive review of multi-speaker word error rate definitions. Given the clarity and extensiveness of their approach, it seemed appropriate for us to follow their methodology. In this definition, underestimation of speakers is penalized by adding empty dummy channels to the hypothesis, whereas overestimation is not penalized. However, in a subsequent paper [28], the authors revised their definition to also penalize overestimation by adding dummy channels to the reference.

In other papers dealing with scenarios involving an unknown number of speakers, the definitions were often ambiguous, or they indicated that redundant hypothesis speakers were discarded. For instance, [6] and [41] mention the removal of redundant speakers, denoting this variant as cpWER-us when dealing with an unknown number of speakers.

In the series of papers on Serialized Output Training by Naoyuki Kanda et al., the initial paper, [42], touches on the problem of having more hypothesis speakers than references but is vague about the exact resolution (referring to the metric as WER, though effectively it aligns with cpWER). Later works in this series did not provide further clarification on this issue.

A.2. Results using the MeetEval cpWER definition

For completeness, we provide the results calculated using the cpWER definition that penalizes overestimation, utilizing the MeetEval toolkit [28]. The inference hyperparameters θ, δ, and Δt were re-optimized on the development dataset using this cpWER definition. Results based on this updated cpWER can be found in Tables 7 and 8 for AMI-SDM and Alimeeting channel 1, respectively. The relative changes in cpWER are consistent with those obtained using the original definition.

Table 7: MeetEval cpWER (%) results on AMI-SDM for various ASR models with speaker attribution (SA) through diarization or separation.

ASR model	SA method	SA system	cpWER(%)				Relative Change
			sub	del	ins	**total**	
Whisper small.en	Diarization	pyannote 3.1	7.6	29.0	4.0	**40.5**	
	Diarization	ToTaToNet	7.8	27.2	2.2	**37.2**	-8.1%
	Separation	ToTaToNet	8.8	24.2	2.4	**35.4**	-12.6%
Whisper medium.en	Diarization	pyannote 3.1	6.7	29.7	3.6	**40.0**	
	Diarization	ToTaToNet	7.0	28.1	2.0	**37.1**	-7.3%
	Separation	ToTaToNet	7.6	24.1	2.2	**33.9**	-15.3%
Whisper large-v2	Diarization	pyannote 3.1	6.4	28.0	3.9	**38.3**	
	Diarization	ToTaToNet	6.8	26.3	2.1	**35.2**	-8.1%
	Separation	ToTaToNet	7.3	22.7	2.6	**32.6**	-14.9%
Nemo conformer large	Diarization	pyannote 3.1	12.0	35.5	2.9	**50.4**	
	Diarization	ToTaToNet	13.2	34.1	1.6	**48.9**	-3.0%
	Separation	ToTaToNet	15.7	23.7	2.0	**41.4**	-17.9%

Table 8: MeetEval cpCER (%) results on Alimeeting channel 1 for various ASR models with speaker attribution (SA) through diarization or separation.

ASR model	SA method	SA system	cpCER(%)				Relative Change
			sub	del	ins	**total**	
Whisper small	Diarization	pyannote 3.1	23.2	35.4	10.0	**68.6**	
	Diarization	ToTaToNet	23.1	35.0	9.6	**67.7**	-1.3%
	Separation	ToTaToNet	18.9	32.4	2.4	**53.7**	-21.7%
Whisper medium	Diarization	pyannote 3.1	18.5	37.9	9.5	**65.9**	
	Diarization	ToTaToNet	18.7	37.3	8.9	**64.9**	-1.5%
	Separation	ToTaToNet	12.5	34.7	1.6	**48.7**	-26.1%
Whisper large-v2	Diarization	pyannote 3.1	17.9	37.9	9.9	**65.6**	
	Diarization	ToTaToNet	18.1	37.3	9.3	**64.7**	-1.4%
	Separation	ToTaToNet	13.0	32.5	1.8	**47.3**	-27.9%

Appendix 3

III

Joonas Kalda, Tanel Alumäe, Martin Lebourdais, Hervé Bredin, Séverin Baroudi, and Ricard Marxer. TalTech-IRIT-LIS speaker and language diarization systems for DISPLACE 2024. In *25th Annual Conference of the International Speech Communication Association, Interspeech 2024*, pages 1635–1639, 2024

TalTech-IRIT-LIS Speaker and Language Diarization Systems for DISPLACE 2024

Joonas Kalda[1], *Tanel Alumäe*[1], *Martin Lebourdais*[2], *Hervé Bredin*[2], *Séverin Baroudi*[3], *Ricard Marxer*[3]

[1]Tallinn University of Technology, Estonia [2]IRIT, Université de Toulouse, CNRS, Toulouse, France
[3]Université de Toulon, Aix Marseille Univ, CNRS, LIS, Toulon, France

firstname.lastname@{taltech.ee, irit.fr, lis-lab.fr}

Abstract

This paper describes the submissions of team TalTech-IRIT-LIS to the DISPLACE 2024 challenge. Our team participated in the speaker diarization and language diarization tracks of the challenge. In the speaker diarization track, our best submission was an ensemble of systems based on the *pyannote.audio* speaker diarization pipeline utilizing powerset training and our recently proposed PixIT method that performs joint diarization and speech separation. We improve upon PixIT by using the separation outputs for speaker embedding extraction. Our ensemble achieved a diarization error rate of 27.1% on the evaluation dataset. In the language diarization track, we fine-tuned a pre-trained Wav2Vec2-BERT language embedding model on in-domain data, and clustered short segments using AHC and VBx, based on similarity scores from LDA/PLDA. This led to a language diarization error rate of 27.6% on the evaluation data. Both results were ranked first in their respective challenge tracks.

Index Terms: DISPLACE 2024, speaker diarization, language diarization

1. Introduction

Speaker diarization is the task of dividing an audio recording into segments based on the speaker identity. The conventional method for tackling this is a multi-stage approach that joins speaker segmentation, local speaker embeddings, and clustering [1]. This approach struggles with overlap-heavy speech, a domain that is better suited for end-to-end neural diarization (EEND) [2, 3]. On the other hand, EEND is data-hungry and has the issue of mispredicting the number of speakers. This has motivated a hybrid approach that replaces the speaker segmentation step of the multi-stage approach with local EEND [4].

Language diarization is the less-studied task of segmenting a recording by the spoken language. It is used as the first step in processing multilingual code-switched speech. Inspired by speaker diarization, both multi-stage [5] and end-to-end neural [6] approaches have been used to solve this task.

The DISPLACE 2024 Challenge is centered on advancing research in the domains of speaker and language diarization, as well as automatic speech recognition (ASR), within multilingual and multi-accent environments [7]. The challenge emphasizes the utilization of realistic speech data, characteristically featuring frequent language switches by speakers at both sentence and phrase levels. DISPLACE 2024 is structured around three evaluation tracks: speaker diarization, language diarization, and ASR.

The dataset for the first two tracks comprises far-field, multi-party multilingual conversational speech recordings, featuring speakers who engage in code-mixing or code-switching across multiple languages. The development set for these tracks consists of 35 recordings, summing up to nearly 20 hours of audio. The evaluation set encompasses 32 recordings, totaling almost 18 hours. Each recorded conversation, lasting around 30 to 60 minutes, involves 3-5 participants fluent in various Indian languages as well as English (with an Indian accent).

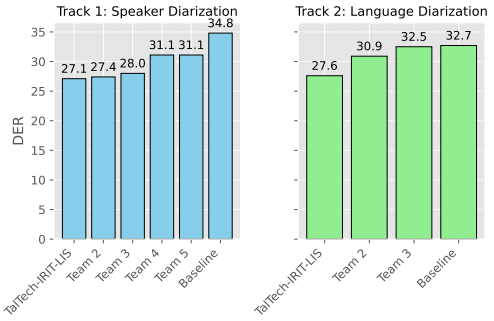

Figure 1: *The results of top-performing teams on DISPLACE 2024 evaluation data.*

For the third track, dedicated to speech recognition, a separate development dataset is provided. This dataset includes 8 recordings, each segmented into single-language regions. The segments are labeled with the corresponding language and accompanied by an orthographic transcript.

Participants in the challenge are permitted to employ any publicly available or proprietary datasets for training and refining their diarization systems. This includes leveraging development data from other tracks within the challenge. These development sets can be utilized for model training and hyperparameter optimization. The performance of systems in Tracks 1 and 2 is evaluated based on the diarization error rate (DER), with overlap and without forgiveness collar.

Our team participated in Tracks 1 and 2 of the challenge. Figure 1 shows that we outperformed other teams in both tracks.

2. Track 1: Speaker diarization

2.1. Methods

2.1.1. Powerset training

Our first standalone system is based on the same approach as the submission #5 of the *pyannote* team at VoxSRC 2023 [8]. This

hybrid approach consists of local end-to-end neural speaker segmentation on a few-second sliding window, neural speaker embedding of each speaker of each window, and agglomerative hierarchical clustering (AHC). The backbone of our local speaker segmentation model is a WavLM-base model [9] pre-trained from scratch on a compound dataset consisting of AISHELL [10], AliMeeting [11], AMI [12], AVA-AVD [13], DIHARD [14], Ego4D [15], MSDWild [16], REPERE [17], and VoxConverse 0.3.0 [18] which is applied on a 10-second sliding window with a stride of 1 second. The 8th layer of this WavLM is fed into an LSTM-based network. The WavLM and the LSTM-based network consist of 94.4M parameters and 2.1M parameters respectively. The training uses powerset multi-label cross-entropy loss [19] with $K_{\max} = 3$ speakers. Speaker embeddings were extracted using the pre-trained ResNet34 model from the WeSpeaker toolkit [20].

2.1.2. PixIT

We also experimented with our recently proposed PixIT method, combining permutation invariant training (PIT) for speaker diarization and mixture invariant training (MixIT) for speech separation [21]. For PixIT, the multitask loss is defined as $\mathcal{L}_{\text{PixIT}} = \lambda\mathcal{L}_{\text{PIT}} + (1-\lambda)\mathcal{L}_{\text{MixIT}}$. It is calculated on pairs of mixtures extracted from the same recording environments so that they contain disjoint sets of speakers and the combined number of speakers is at most $K_{\max}$. The mixtures are added together to create mixtures of mixtures (MoMs). $\mathcal{L}_{\text{MixIT}}$ is calculated only on the MoMs while $\mathcal{L}_{\text{PIT}}$ also utilizes the original mixtures. The local joint model is based on the TasNet architecture [22]. The feature encoder concatenates the outputs of the pre-trained WavLM-large [9] model and a 1-D convolutional encoder. The masking network outputs $K_{\max}$ masks which are then independently processed by either a 1-D convolutional decoder or a fully connected neural network for local speech separation or speaker diarization respectively.

To perform global speaker diarization, the speaker diarization branch of the local joint model is used in the same pipeline as in Section 2.1.1. The only difference is that for speaker embeddings we used a pre-trained ECAPA-TDNN model [23] available in [24].

We experimented with multiple improvements to the original PixIT system. First, we utilized separated sources output by the joint model for speaker embedding extraction instead of the original audio. This allows for additional information from the overlapped regions and further integrates the two tasks. A potential downside is that separation outputs can include artifacts the speaker embedding model has not seen during its training. Second, we used a DPTNet [25] instead of a DPRNN [26] as the masking network which was shown to perform better at speech separation albeit on synthetic data. We kept the hyperparameters the same as in the original work. Finally, to improve the quality of the local speaker embeddings we increased the length of the sliding window from 5 to 10 seconds while increasing the stride of the convolutional encoder two-fold.

The total number of parameters for the PixIT model is 319M when using a DPRNN and 324M when using a DPTNet.

2.2. Results

For fine-tuning our speaker diarization systems, we divided the DISPLACE 2024 development set further into train and development splits with the latter containing the recordings M030, B022, M019, and B034. Accordingly, we will only report results on the evaluation dataset of the challenge.

Table 1: *DERs (%) obtained on Track 1 evaluation data for different configurations of the PixIT method. $*$ denotes submissions made during the post-evaluation phase of the competition.*

Submission		Eval
	DISPLACE 2024 baseline	34.76
#1	Original PixIT system with $\lambda = 0.1$	30.05
#2	#1 + embeddings from separated sources	29.44
#3	#2 + DPTNet as the masking network	27.15^{*}
#4	#3 + 10s sliding window	26.70^{*}

Table 2: *DERs (%) obtained on Track 1 evaluation dataset for different system configurations. Our best-performing system for Phase 1 of the competition is in bold.*

Submission		Eval
	DISPLACE 2024 baseline	34.76
#2	PixIT	29.44
#5	powerset off-the-shelf	30.57
#6	powerset fine-tuned	27.34
#7	powerset fine-tuned, *max_speakers* $= 5$	29.09
#8	powerset fine-tuned, *max_speakers* $= 6$	28.35
#9	powerset fine-tuned, *max_speakers* $= 7$	27.29
#10	DOVER-Lap of #2, #6 and #9	27.27
#11	DOVER-Lap of #2, #6, #7, #8 and #9	27.12
#12	DOVER-Lap of #2, #6 and #7	**27.08**

The performance on the evaluation dataset for our PixIT-based systems is detailed in Table 1. Optimizing for the DER on the development data, we found $\lambda = 0.1$ to perform the best. Using the separated sources predicted by the joint model for extracting speaker embeddings instead of the original audio yields an improvement in DER from 30.1% to 29.4%. This shows that the additional information extracted from the overlapped regions outweighs the negative effect of the presence of artifacts in the separated sources. An additional 7.8% relative improvement is achieved by replacing the DPRNN with a DPTNet as the masking network. The superior performance of DPTNet thus extends to the case of shared training on real data. Lastly, extending the sliding window length to 10 seconds further improves DER by a relative 1.7%.

The results of our systems using powerset training and ensemble methods are shown in Table 2. Fine-tuning the powerset system allowed us to get from 30.6% down to 27.3% DER on the evaluation data. We also experimented with constraining the maximal number of speakers in clustering to either 5, 6, or 7. The last case yields slight improvements to DER while others perform worse than the unconstrained system. Finally, we use greedy DOVER-Lap [27] to combine the PixIT system with various powerset systems. We found the best results from choosing the unconstrained fine-tuned version and the fine-tuned version constrained to a maximum of 5 speakers. This is likely because the variation in outputs is the greatest for that pair of systems.

2.3. Runtime performance

The powerset system was fine-tuned using a single V100 GPU for approximately 1h. On the same hardware, it takes 10m30s to process the DISPLACE 2024 evaluation set. PixIT systems were trained on a single 80GB A100 GPU for approximately 3 days. It takes 1.2 hours for these systems to process the evaluation dataset.

3. Track 2: Language diarization

3.1. Methods

In the language diarization track, we used the more conventional diarization technique, consisting of speech detection, segmentation into short overlapping windows, extraction of segment embeddings, and clustering of the segments, with VBx [1] based refinement of the initial clustering hypothesis.

As the first step in processing target speech data, segments containing speech were found from the recordings, using the Silero VAD model [28]. Speech segments were further subsegmented, using a 5-second window with a 1-second shift. The use of 5-second window was inspired by the results from DISPLACE 2023 [29] and verified by our own initial experiments.

The resulting 5-second segments were processed by the language embedding model, which produces a 512-dimensional vector for each short segment. The backbone of the embeddings extractor is the Wav2Vec2-BERT model[1] shared by the Seamless4MT project [30]. This model was pre-trained on 4.5M hours of unlabeled audio data covering more than 143 languages, using self-supervised loss. Wav2Vec2-BERT follows the same architecture as Wav2Vec2.0 [31], but replaces the attention-block with a Conformer-block as introduced in [32]. It also uses mel-spectrogram representation of the audio as input, instead of the raw waveform. This particular Wav2Vec2-BERT model comprises 24 Conformer layers with approximately 600M parameters. The Wav2Vec2-BERT model was converted into a language identification model by feeding its outputs through an attentive pooling layer, a fully connected layer with ReLU and BatchNorm, and the final output layer, corresponding to the languages of the training set. The model is trained using cross-entropy loss on random 2 to 4-second chunks of language-labeled training data. Point source noises and simulated room impulse responses (RIRs) from the SLR28 Room Impulse Response and Noise Database [33] were used for on-the-fly data augmentation. Segment embeddings are extracted from the output of the first dense layer after the pooling layer. Low-rank adaptation (LoRA) [34] is used for finetuning the pre-trained Wav2Vec2-BERT model, with $\texttt{rank} = 32$, $\alpha = 32$ and $\texttt{dropout} = 0.05$. Supervised training was performed using an effective batch size of 64, peak learning rate 10^{-3} and weight decay 10^{-3}. Due to the use of LoRA, the number of trainable parameters in the model is only 7.9M.

We tried various datasets for training the language embedding model. Initial experiments with the VoxLingua107 dataset [35] gave poor results on DISPLACE data (see section 3.3). Therefore, we opted to use data from NIST Language Recognition Evaluations (LREs) and Speaker Recognition Evaluations (SREs) for training the embedding model. Specifically, the language embeddings extractor was trained on NIST LRE 2003 evaluation data (LDC2006S31), NIST LRE 2005 evaluation data (LDC2008S05), NIST LRE 2007 evaluation data (LDC2009S04), NIST LRE 2009 evaluation data (LDC2014S06), NIST LRE 2007 training data (LDC2009S05), NIST SRE 2008 training data (LDC2011S05). Those datasets contain mostly conversational telephone speech, including English with Indian accent. The amount of speech data per language is given in Table 3.

Although the languages used in DISPLACE 2024 Track 2 development and evaluation data were not known during the challenge period, the DISPLACE 2023 [36] report suggests that they could include Indian-accented English, Hindi, Telugu, Bangla/Bengali, Kannada, Tamil. Table 3 shows that Telugu and Kannada were not covered by the training data used for training language embeddings. In order to adapt the embeddings to the DISPLACE 2024 scenario, we fine-tuned the embeddings model on development data from Track 3 which has been segmented and transcribed according to the language. This gives us around 3.5 hours of in-domain data (see Table 4). Fine-tuning was performed for 6 epochs from the checkpoint trained on 10 epochs of NIST data, using a learning rate schedule where the peak learning rate is 10 times smaller than when training the initial model.

The 5-second segments were clustered using a language recognition model based on a LDA/PLDA, trained on Track 3 development data. The LDA/PLDA model transforms centered language embeddings to 150 dimensions using LDA and estimates a PLDA model on the length-normalized features. The LDA/PLDA model is used to evaluate the cross-similarity across all 5-second segment pairs in each target recording. The similarities are used to perform initial clustering of the 5-second segments, using AHC. The initial language segmentation is finally refined using Bayesian HMM clustering (VBx) [1], using the following parameters: $P_{\texttt{loop}} = 0.9$, $F_a = 9$, $F_b = 4$.

Table 3: *Amount of training data per language for training the language embedding model for Track 2.*

Language	Hours	Language	Hours	Language	Hours
Amharic	5.4	Haiti Creole	4.4	Russian	30.4
Arabic	13.4	Hausa	5.3	Spanish	26.0
Azerbaijani	5.0	Hindi	32.9	Swahili	5.4
Belarusian	4.9	Indonesian	1.7	Tagalog	6.0
Bengali	9.8	Italian	4.6	Tamil	8.6
Bosnian	4.8	Japanese	27.1	Thai	35.2
Bulgarian	5.1	Khmer	0.1	Tibetan	5.0
Cantonese	7.6	Korean	27.6	Tigrinya	0.0
Chinese	113.9	Lao	0.1	Turkish	5.4
Croatian	5.1	Pashto	5.4	Ukrainian	5.3
English	646.3	Persian	18.0	Urdu	11.6
French	9.3	Portuguese	5.4	Uzbek	5.8
Georgian	5.5	Punjabi	0.7	Vietnamese	29.5
German	5.6	Romanian	5.4		

Table 4: *Amount of data per language in Track 3 development data.*

Language	Amount (hh:mm)
Bengali	0:26
Hindi	0:24
English	1:47
Kannada	0:12
Telugu	0:37

3.2. Results

Table 5 presents the performance of various baseline systems and our own models on the development and evaluation datasets for Track 2. Confidence intervals [37] on development data are computed by treating each recording as IID. Notably, the DISPLACE 2023 baseline, which uses an EPACA-TDNN model trained on VoxLingua107 dataset for generating language embeddings, followed by the clustering of short segments using

[1] https://huggingface.co/facebook/w2v-bert-2.0

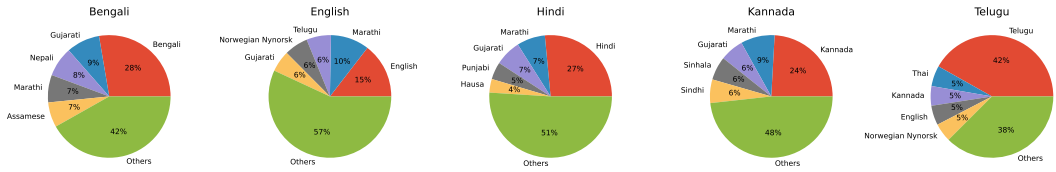

Figure 2: *Top 5 most frequent predicted languages for Track 3 development utterances in the five given languages, based on the language identification model trained on VoxLingua107.*

Table 5: *Language diarization error rates (DER) on Track 2 development and evaluation data, using different model configurations.*

Training data for embeddings	LDA/PDA	VBx	Dev (conf. int.)	Eval
Baselines				
No segmentation			44.4 (41.9-49.1)	
DISPLACE 2023 baseline			48.6 (46.3-52.8)	
DISPLACE 2024 baseline			40.7 (37.9-45.3)	32.7
VL107	VL107	✗	38.3 (35.7-43.4)	
VL107	Track3 dev	✗	32.9 (30.3-37.5)	
VL107 + Track3 dev	Track3 dev	✗	30.1 (27.7-34.8)	
NIST	NIST	✗	30.9 (28.4-35.6)	
NIST	NIST	✓	29.7 (27.6-34.9)	
NIST	Track3 dev	✗	31.3 (28.9-36.3)	
NIST	Track3 dev	✓	28.7 (26.1-33.5)	29.6
NIST + Track3 dev	Track3 dev	✗	29.3 (26.8-34.2)	
NIST + Track3 dev	Track3 dev	✓	28.2 (25.6-33.0)	**27.6**

AHC, does not outperform the simplistic baseline that attributes all speech to a single language. However, the DISPLACE 2024 baseline that substitutes the EPACA-TDNN language embeddings with language detection posterior probabilities derived from Whisper, and incorporates VBx into the clustering step, achieves an improvement over the "uninformative" baseline.

The results further indicate that language embeddings trained using data from NIST LREs and SREs significantly outperform those trained with VoxLingua107 (VL107) data for the DISPLACE 2024 dataset. However, substantial gains are observed when in-domain data from Track 3 is utilized for estimating the LDA/PLDA model and for finetuning the embeddings. This approach not only enhances the performance of the VoxLingua107 based model but also narrows the gap to the models trained on NIST datasets. The system corresponding to the last line in the table obtained the best results on evaluation data among all teams.

3.3. Analysis

Our investigation revealed that the VoxLingua107 dataset, effective for various language recognition tasks, showed weak performance on the DISPLACE 2024 dataset. To decipher the underlying causes of this problem, we assessed the language identification capabilities of a model trained on VoxLingua107 using the Track 3 development dataset, evaluating it through its posterior probabilities without employing LDA/PLDA postprocessing. Although the model achieved an accuracy of 95.4% on the VoxLingua107 development dataset, its performance dramatically decreased to 22.2% on the Track 3 dataset. Our analysis, shown in Figure 2, identified a trend across languages: while the correct language was often identified, recall rates were significantly low, from 15% for English to 42% for Telugu. This drop in accuracy can be attributed to factors like environmental noise and the conversational speech style. However, a major reason for the decline was the inclusion of non-native speech in the DISPLACE 2024 data. Prior study has shown that models trained on VoxLingua107 face dramatic accuracy losses with non-native accents [38], and that such models could be improved by also using a lexicon-free character-based speech recognition for various languages to transcribe speech, followed by applying a text-based classification model on these transcripts. The combined model approach could potentially enhance language diarization and segmentation tasks as well.

3.4. Runtime performance

Training of the language embedding model was performed on 6 P100 GPUs and it took approximately 4 hours. Finetuning the model on Track 3 data takes a few minutes on one GPU. Processing test data from start to finish takes about 0.08 $\times$ realtime, assuming one GPU and one CPU.

4. Conclusion

This work presents our submissions to the DISPLACE 2024 challenge. For the speaker diarization track, our best system combines *pyannote.audio* speaker diarization pipelines where the segmentation is done either by a model trained with a powerset objective function or by a joint separation-diarization model trained with our recently proposed PixIT loss. The latter system is improved upon by extracting speaker embeddings directly from local separated sources. The ensemble reaches a DER of 27.1% on the phase one evaluation data. In the language diarization track, a 27.6% DER score is achieved by combining local language embeddings from a pre-trained Wav2Vec2-BERT model with clustering using AHC and VBx, based on similarity scores from LDA/PLDA. Our systems achieved first places in both of the tracks.

5. Acknowledgements

The research reported in this paper was supported by the Agence de l'Innovation Défense under the grant number 2022 65 0079, and by the Estonian Centre of Excellence in AI. This work was granted access to the HPC resources of GENCI-IDRIS under the allocations AD011014274.

6. References

[1] F. Landini, J. Profant, M. Diez, and L. Burget, "Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: theory, implementation and analysis on standard tasks," *Computer Speech & Language*, vol. 71, p. 101254, 2022.

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

[3] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with self-attention," in *ASRU*, 2019.

[4] 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.

[5] S. Baghel, S. Ramoji, S. Jain, P. R. Chowdhuri, P. Singh, D. Vijayasenan, and S. Ganapathy, "Summary of the DISPLACE challenge 2023–diarization of speaker and language in conversational environments," *arXiv preprint arXiv:2311.12564*, 2023.

[6] J. Mishra, A. Agarwal, and S. M. Prasanna, "Spoken language diarization using an attention based neural network," in *NCC*, 2021.

[7] S. B. Kalluri, P. Singh, P. R. Chowdhuri, A. Kulkarni, S. Baghel, P. Hegde, S. Sontakke, D. K. T, S. R. M. Prasanna, D. Vijayasenan *et al.*, "The Second DISPLACE Challenge : DIarization of SPeaker and LAnguage in Conversational Environments," in *Interspeech*, 2024.

[8] S. Baroudi, H. Bredin, A. Plaquet, and T. Pellegrini, "pyannote.audio speaker diarization pipeline at VoxSRC 2023," http://mm.kaist.ac.kr/datasets/voxceleb/voxsrc/data_workshop_2023/reports/pyannote_report.pdf, Tech. Rep., 2023.

[9] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[10] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "AIShell-1: An open-source Mandarin speech corpus and a speech recognition baseline," in *Oriental COCOSDA*, 2017.

[11] F. Yu, S. Zhang, Y. Fu, L. Xie, S. Zheng, Z. Du, W. Huang, P. Guo, Z. Yan, B. Ma *et al.*, "M2Met: The ICASSP 2022 multi-channel multi-party meeting transcription challenge," in *ICASSP*, 2022.

[12] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal *et al.*, "The AMI meeting corpus: A pre-announcement," in *ICMI*, 2005.

[13] E. Z. Xu, Z. Song, S. Tsutsui, C. Feng, M. Ye, and M. Z. Shou, "AVA-AVD: Audio-visual speaker diarization in the wild," in *ACM MM*, 2022.

[14] N. Ryant, P. Singh, V. Krishnamohan, R. Varma, K. Church, C. Cieri, J. Du, S. Ganapathy, and M. Liberman, "The third dihard diarization challenge," *arXiv preprint arXiv:2012.01477*, 2020.

[15] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu *et al.*, "Ego4d: Around the world in 3,000 hours of egocentric video," in *CVPR*, 2022.

[16] T. Liu, S. Fan, X. Xiang, H. Song, S. Lin, J. Sun, T. Han, S. Chen, B. Yao, S. Liu *et al.*, "MSDWild: Multi-modal speaker diarization dataset in the wild." in *Interspeech*, 2022.

[17] A. Giraudel, M. Carré, V. Mapelli, J. Kahn, O. Galibert, and L. Quintard, "The REPERE Corpus : a multimodal corpus for person recognition," in *LREC'12*, 2012.

[18] J. S. Chung, J. Huh, A. Nagrani, T. Afouras, and A. Zisserman, "Spot the conversation: Speaker diarisation in the wild," in *Interspeech*, 2020.

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

[20] H. Wang, C. Liang, S. Wang, Z. Chen, B. Zhang, X. Xiang, Y. Deng, and Y. Qian, "Wespeaker: A research and production oriented speaker embedding learning toolkit," in *ICASSP*, 2023.

[21] J. Kalda, C. Pagés, R. Marxer, T. Alumäe, and H. Bredin, "PixIT: Joint training of speaker diarization and speech separation from real-world multi-speaker recordings," *arXiv preprint arXivL:2403.02288*, 2024.

[22] Y. Luo and N. Mesgarani, "TasNet: Time-domain audio separation network for real-time, single-channel speech separation," in *ICASSP*, 2018.

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

[24] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong *et al.*, "Speechbrain: A general-purpose speech toolkit," *arXiv preprint arXiv:2106.04624*, 2021.

[25] J. Chen, Q. Mao, and D. Liu, "Dual-Path Transformer Network: Direct Context-Aware Modeling for End-to-End Monaural Speech Separation," in *Interspeech*, 2020, pp. 2642–2646.

[26] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path RNN: Efficient long sequence modeling for time-domain single-channel speech separation," in *ICASSP*, 2020.

[27] D.Raj, P.Garcia, Z.Huang, S.Watanabe, D.Povey, A.Stolcke, and S.Khudanpur, "DOVER-Lap: A method for combining overlap-aware diarization outputs," in *SLT*, 2021.

[28] Silero Team, "Silero VAD: pre-trained enterprise-grade voice activity detector (VAD), number detector and language classifier," https://github.com/snakers4/silero-vad, 2021.

[29] B. Vachhani, D. Singh, and R. Lawyer, "Multi-resolution approach to identification of spoken languages and to improve overall language diarization system using Whisper model," in *Interspeech*, 2023.

[30] Seamless Communication, L. Barrault, Y.-A. Chung, M. C. Meglioli, D. Dale, N. Dong, M. Duppenthaler, P.-A. Duquenne, B. Ellis, H. Elsahar *et al.*, "Seamless: Multilingual expressive and streaming speech translation," *arXiv preprint arXiv:2312.05187*, 2023.

[31] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *NeurIPS*, 2020.

[32] 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.

[33] 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 *ICASSP*, 2017.

[34] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[35] J. Valk and T. Alumäe, "VoxLingua107: a dataset for spoken language recognition," in *SLT*, 2021.

[36] S. Baghel, S. Ramoji, Sidharth, R. H, P. Singh, S. Jain, P. Roy Chowdhuri, K. Kulkarni, S. Padhi, D. Vijayasenan *et al.*, "The DISPLACE Challenge 2023 - DIarization of SPeaker and LAnguage in Conversational Environments," in *Interspeech*, 2023.

[37] L. Ferrer and P. Riera, "Confidence intervals for evaluation in machine learning." [Online]. Available: https://github.com/luferrer/ConfidenceIntervals

[38] K. Kukk and T. Alumäe, "Improving language identification of accented speech," in *Interspeech*, 2022.

Appendix 4

IV

Joonas Kalda, Séverin Baroudi, Martin Lebourdais, Clément Pagés, Ricard Marxer, Tanel Alumäe, and Hervé Bredin. Design choices for PixIT-based speaker-attributed ASR: Team ToTaTo at the NOTSOFAR-1 challenge. *Computer Speech & Language*, page 101824, 2026

Design Choices for PixIT-based Speaker-Attributed ASR: Team ToTaTo at the NOTSOFAR-1 Challenge

Joonas Kalda, Séverin Baroudi, Martin Lebourdais, Clément Pagés, Ricard Marxer, Tanel Alumäe, Hervé Bredin

Design Choices for PixIT-based Speaker-Attributed ASR: Team ToTaTo at the NOTSOFAR-1 Challenge

Joonas Kalda[a], Séverin Baroudi[b], Martin Lebourdais[c], Clément Pagés[c], Ricard Marxer[b], Tanel Alumäe[a], Hervé Bredin[c]

[a] *Tallinn University of Technology, Tallinn, Estonia*
[b] *Université de Toulon, Aix Marseille Univ, CNRS, LIS, Toulon, France*
[c] *IRIT, Université de Toulouse, CNRS, Toulouse, France*

Abstract

PixIT is a recently proposed joint training framework that integrates Permutation Invariant Training (PIT) for speaker diarization and Mixture Invariant Training (MixIT) for speech separation. By leveraging diarization labels, PixIT addresses MixIT's limitations, producing aligned sources and speaker activations that enable automatic long-form separation. We investigate applications of PixIT on the speaker-attributed automatic speech recognition (SA-ASR) task based on our systems for the NOTSOFAR-1 Challenge. We explore modifications to the joint ToTaToNet by integrating advanced self-supervised learning (SSL) features and masking networks. We show that fine-tuning an ASR system on PixIT-separated sources significantly boosts downstream SA-ASR performance, outperforming standard diarization-based baselines without relying on synthetic data. We explore lightweight post-processing heuristics for improving SA-ASR timestamp errors caused by long silences and artifacts present in file-level separated sources. We also show the potential of extracting speaker embeddings for the diarization pipeline directly from separated sources, with performance rivaling standard methods without any fine-tuning of speaker embeddings. On the NOTSOFAR-1 Challenge dataset, our PixIT-based approach outperforms the CSS-based baseline by 20% in terms of tcpWER after fine-tuning the ASR system on the separated sources. Notably, even when using the same ASR model as the baseline, our system is able to outperform it, without using any of the provided domain-specific synthetic data. These advancements position PixIT as a robust and flexible solution for real-world SA-ASR.

Keywords: Speech Separation, Speaker Diarization, Speaker-Attributed

Automatic Speech Recognition, Speaker Embeddings, SSL Models, Joint Training

1. Introduction

Speech separation, also known as the cocktail party problem, involves isolating individual speakers' voices from a mixed audio signal. The predominant approaches for training deep-learning models for this task have been based on supervised training, using clean isolated speaker signals as training labels [1, 2, 3]. Such signals are unavailable in real-world scenarios. Even when using close-distance microphones, recorded signals typically contain cross-talk from other speakers, making truly isolated recordings difficult to obtain. Thus synthetic data has to be used for supervised training where two or more independently recorded clean speaker signals are added to create an artificial mixture. However, this approach results in a domain mismatch when applied to real-world scenarios. This is not an issue for unsupervised approaches that do not rely on the existence of clean ground-truth signals. A variety of methods have been proposed for this, utilizing multi-channel audio, which allows to estimate spatial locations of sources [4, 5]. A more general approach that also works for single-channel training data is Mixture Invariant Training (MixIT) [6]. MixIT operates by combining two mixtures from the target domain to create a "mixture of mixtures" (MoM). The separation model is then trained to estimate source signals so that they can be combined to recreate the original mixtures. An obvious limitation of MixIT is that the number of speakers in a MoM is twice that of a single mixture. This means that the separation model now faces a different kind of domain mismatch problem. The number of outputs for the separation model is doubled and as a result, the model struggles with over-separation when applied to real-world conversations – a single speaker's signal can end up divided across multiple predicted sources. Combining the use of MixIT and traditional supervised separation training has been shown to reduce this problem [7] at the cost of re-introducing reliance on synthetic data.

Besides domain mismatch, speech separation struggles with inference on long-form audio. Speech separation networks are trained on short mixtures and inference on long recordings requires some way of stitching together local outputs. The predominant approach for this has been continuous speech separation (CSS) [8]. Separation outputs on overlapping sliding windows

are stitched together based on a similarity metric that is calculated on the overlapped region. However, when there is a longer pause between utterances of a particular speaker, these utterances can end up in different channels. Thus for long-form separation, CSS has to be accompanied by a speaker diarization system that predicts which speaker each utterance corresponds to.

Speaker diarization is the process of identifying who speaks when in an audio recording. Traditional approaches follow a multi-step process: first using voice activity detection (VAD) to locate speech segments, then extracting speaker-specific voice characteristics (embeddings), and finally applying clustering algorithms to group similar voices together [9, 10]. While often effective, these methods struggle with overlapping speech segments. End-to-end neural diarization (EEND) has been introduced to address overlapping speech by directly modeling speaker activities with a single network [11, 12]. However, EEND requires substantial training data and can struggle with estimating the number of speakers. To leverage the strengths of both approaches, a hybrid framework has been developed. This method applies EEND to shorter audio segments and subsequently integrates the results using speaker embeddings and clustering techniques [13, 14].

Speech separation and speaker diarization systems frequently work together, particularly for CSS applications. Applying the diarization system to already separated speech sources solves the overlapped speech problem, thus enabling the use of multi-step approaches [15]. However, this arrangement has a drawback: any errors or artifacts introduced during the speech separation process will affect the subsequent diarization results. The two systems can also be arranged in the reverse order, with diarization preceding separation. In this configuration, the speech separation model benefits from the diarization system's output of time-domain speaker masks, which provide a natural foundation for time-frequency domain masking. This interdependence highlights how the two tasks complement each other, as they both extract speaker-specific information but at different levels of detail — separation focusing on the acoustic signal level and diarization on the speaker identity level.

These observations have motivated a suite of joint training approaches. The Recurrent Selective Attention Network (RSAN) architecture [16] was the first all-neural model to jointly perform speech separation, speaker diarization, and speaker counting. RSAN processes audio iteratively using sliding blocks, extracting speakers by estimating masks for each, guided by

embeddings and residual masks from previous blocks and iterations. Another joint approach is the End-to-End Neural Diarization and Speech Separation (EEND-SS) architecture [17]. Built on the EEND framework for diarization and speaker counting, and Conv-TasNet [1] for separation, EEND-SS refines separation using diarization outputs. This includes leveraging the estimated number of speakers and probabilities of speech activity to enhance separation. While these joint methods offer promising results, they still rely on synthetic data for training the separation components. A joint training method on real-world data was proposed in [18] but is restricted to multi-channel audio.

PixIT [19] was proposed as a joint training approach for speaker diarization and speech separation that builds on MixIT by incorporating PIT for diarization. It solves the main problems of MixIT while imposing the small requirement of speaker diarization labels being available for training. A key advantage of models trained with PixIT is that they produce sources aligned with speaker sources, meaning that when local diarization outputs are stitched following the best-of-both-worlds approach, the long-form separated audio sources are automatically obtained. The effectiveness of PixIT has been demonstrated in practical applications. It was shown to improve the performance of off-the-shelf ASR systems on single-channel meeting recordings substantially over the standard speaker attribution baseline that does not rely on synthetic data, where the speaker diarization model is used to divide speech segments between speakers. Further validation came through our DISPLACE challenge submission [20], which revealed an additional benefit of this joint training approach: using locally separated sources instead of the original audio for extracting speaker embeddings can improve diarization performance.

In this work we extend our submission [20] to the Natural Office Talkers in Settings of Far-field Audio Recordings (NOTSOFAR-1) Challenge [21]. The NOTSOFAR-1 Challenge addresses the SA-ASR task in diverse and realistic meeting scenarios, featuring both single-channel and known-geometry multi-channel tracks. PixIT was originally introduced for single-channel scenarios and thus our participation focused exclusively on this track. Leveraging PixIT in a multi-channel context remains a topic for future work.

We extend prior efforts by exploring new architectures for the joint TOTATONET model, aiming to improve separation performance. We demonstrate that fine-tuning an ASR system on PixIT-separated sources significantly boosts downstream speaker-attributed ASR (SA-ASR) performance. We show the further potential of PixIT in intertwining the separation and

diarization tasks by analyzing the use of separated sources for the clustering step of speaker diarization. Finally, we validate PixIT's generalizability by applying it to the NOTSOFAR-1 dataset, demonstrating that it can outperform CSS even in the presence of domain-specific synthetic data.

The main contributions of this work are as follows:

- Exploring the use and impact of alternative self-supervised learning (SSL) features and masking networks to improve for the ToTaToNet.
- Conducting an in-depth analysis of speaker embedding extraction from separated sources across multiple datasets and varying overlap percentages.
- Improving downstream SA-ASR performance by fine-tuning the ASR model on PixIT-separated sources and developing a post-processing heuristic for timestamps.
- Improving our system for the NOTSOFAR-1 Challenge, demonstrating the effectiveness of PixIT in comparison to CSS without using the provided domain-specific synthetic data.
- Open-sourcing the recipes for the above[1].

2. Methodology

2.1. PixIT

PixIT employs a joint modeling approach that produces both speaker activations and aligned source signals. Specifically, the i^{th} predicted speaker activation corresponds to the same speaker as the i^{th} separated source for all values of i. Our joint ToTaToNet model is shown in Figure 1. It builds on a standard TasNet architecture [22]. We enhance the 1-D encoder by concatenating SSL features to enrich the input representation. To enforce alignment, the diarization branch operates directly on the separated encoded sources, processing each source independently similarly to a VAD system [19].

Supervised speaker diarization is trained on samples consisting of an audio segment $\mathbf{X}$ and the corresponding speaker activity labels $\mathbf{y} \in \{0, 1\}^{K_{\max} \times T}$,

[1] `https://github.com/joonaskalda/PixIT-design-choices`

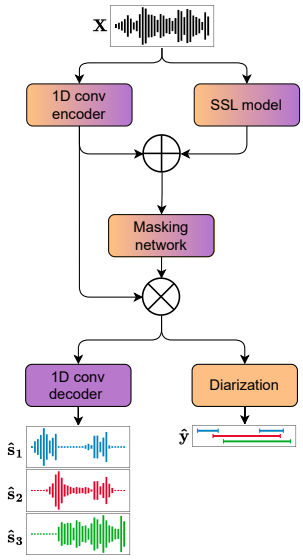

Figure 1: The ToTaToNet architecture for joint speaker diarization and speech separation.

where $y_{k,t} = 1$ indicates speaker k is active at time frame t and $K_{\max}$ is the maximum number of speakers per segment. The standard PIT-loss is,

$$\mathcal{L}_{\text{PIT}}(\mathbf{y}, \hat{\mathbf{y}}) = \min_{\mathbf{P}} \sum_{k=1}^{K_{\max}} \mathcal{L}_{\text{BCE}}\left(\mathbf{y}_k, [\mathbf{P}\hat{\mathbf{y}}]_k\right),$$

where $\hat{\mathbf{y}} \in [0,1]^{K_{\max} \times T}$ denotes the predicted activities, $\mathbf{P} \in \{0,1\}^{K_{\max} \times K_{\max}}$ the permutation matrix, and $\mathcal{L}_{\text{BCE}}$ is the standard binary cross-entropy loss.

For MoMs we sample two non-overlapping audio chunks $(\mathbf{X}^1, \mathbf{y}^1)$ and $(\mathbf{X}^2, \mathbf{y}^2)$ with distinct speaker sets, ensuring the total number of speakers does not exceed $K_{\max}$. This restriction mitigates MixIT's over-separation issue. The mixture $\mathbf{X}^{\text{MoM}}$ is formed by summing the chunks: $\mathbf{X}^{\text{MoM}} = \mathbf{X}^1 + \mathbf{X}^2$. The corresponding labels are concatenated as $y^{\text{MoM}}_{:,t} = (y^1_{:,t}, y^2_{:,t})$, removing

inactive speaker rows to maintain $\mathbf{y}^{\text{MoM}} \in \{0,1\}^{K_{\max} \times T}$.

The MixIT loss is defined as:

$$\mathcal{L}_{\text{MixIT}}(\{\mathbf{X}_n\}, \hat{\mathbf{s}}) = \min_{\mathbf{A}} \sum_{n=1}^{2} \mathcal{L}_{\text{SI-SDR}}\left(\mathbf{X}_n, [\mathbf{A}\hat{\mathbf{s}}]_n\right),$$

where $\hat{\mathbf{s}} \in \mathbb{R}^{M \times T}$ are the separated sources, $\mathbf{A} \in \{0,1\}^{2 \times M}$ is a mixing matrix with each column summing to one, and $\mathcal{L}_{\text{SI-SDR}}$ is the negative scale-invariant signal-to-distortion ratio loss [23]. Limiting the number of speakers allows using a smaller $M = K_{\max}$.

The overall multi-task loss combines PIT and MixIT losses:

$$\begin{aligned}\mathcal{L}_{\text{PixIT}} = \lambda\Big(&\mathcal{L}_{\text{PIT}}(\mathbf{y}^1, \hat{\mathbf{y}}^1) + \mathcal{L}_{\text{PIT}}(\mathbf{y}^2, \hat{\mathbf{y}}^2) + \mathcal{L}_{\text{PIT}}(\mathbf{y}^{\text{MoM}}, \hat{\mathbf{y}}^{\text{MoM}})\Big) \\ &+ (1-\lambda)\mathcal{L}_{\text{MixIT}}\left(\{\mathbf{X}_n\}, \hat{\mathbf{s}}\right)\end{aligned}$$

where $\lambda = 0.5$ was empirically chosen.

2.2. SSL features

A basic ToTaToNet architecture relies on WavLM [24] (LARGE version), a self-supervised model pre-trained beforehand on raw unlabeled audio, and subsequently employed to generate meaningful representations that are used by PixIT to solve both diarization and separation in an end-to-end fashion. The SSL model ingests audio that is processed through a series of convolutional layers (CNN) which extract low-level acoustic features. After randomly masking parts of these features, the representations are passed through a series of Transformer encoders which model long-range dependencies and capture contextualized informations. The task of the SSL model is to predict the hidden units (pseudo-labels) related to the masked portions of the Transformer input. To generate these labels, waveforms are first processed, either through MFCCs or from another SSL model, to generate features that are discretized to produce hidden units (or tokens) using k-means clustering.

Deriving from Masked Language Modeling (BERT [25]) applied to audio (HuBERT [26]), WavLM attempts to specialize itself towards speaker-identity related tasks, and overlapping speech scenarios by introducing an utterance/noise mixing strategy to the input audio. While the pseudo-labels to predict are those of the clean utterance, the model implicitly learns to

de-noise the input in a self-supervised manner. By ingesting noisy and/or overlapping utterances, the SSL model becomes more robust in handling such scenarios, making it particularly beneficial for tasks such as speaker diarization or speech separation, which are of equal importance for PixIT.

To further improve upon the current ToTaToNet (and study the impact of SSL features in said architecture), we replace the current WavLM model with the open-source Conformer-based W2v-BERT 2.0 [27] speech encoder[2] from the Seamless project [28]. While WavLM uses a Masked Language Modeling (MLM) loss only, W2v-BERT 2.0 combines it with a contrastive loss derived from wav2vec2.0 [29]. The pseudo-labels are generated from a quantization block that produces both the target-context vectors (used to compute the contrastive loss over a first series of 12 layers), and the hidden units for the MLM loss (which concerns the 12 last layers). As a result, W2v-BERT 2.0 possesses a much larger architecture than WavLM, featuring 24 Conformer layers [30] (580M parameters), as opposed to the 24 Transformer layers from WavLM-LARGE (315M parameters). The pre-training dataset is also significantly larger for W2v-BERT 2.0 (4.5 millions of hours, and 143 different languages), as opposed to WavLM-LARGE (96k hours of English-only content assembled from LibriLight, VoxPopuli, GigaSpeech). For the remainder of our study, we will use WavLM referring to the LARGE version.

2.3. Masking networks

The masking network employed by ToTaToNet is the Dual-Path Recurrent Neural Network (DPRNN) [2]. Following the TasNet principle [22], the features Z extracted from the encoder are segmented into S overlapping chunks of fixed size K and overlap P (as depicted in Figure 2) :

$$Z \in \mathbb{R}^{N \times L} \Rightarrow Z_{b=1} \in \mathbb{R}^{N \times K \times S} \tag{1}$$

where N represents the feature size, L represents the total number of frames resulting from the encoder, and b being the index of the current layer ($Z_{b=1}$ refers to the input of the first layer and the output of the segmentation step).

After concatenating each chunk, the DPRNN ends up with an effective 3D-tensor Z_b of dimensions $N \times K \times S$. The latter is processed and normalized

[2]https://huggingface.co/facebook/w2v-bert-2.0

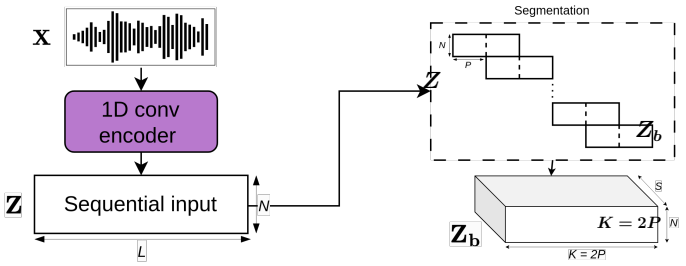

Figure 2: Segmentation process of the DPRNN masking network (figure extracted from [2]).

by intra-chunk (f_b) and inter-chunk (h_b) BiLSTMs respectively, for a specific amount of layers (as summarized in Figure 3):

$$U_b = f_b(Z_b[:, :, i]) \quad \forall i \in \{1, \ldots, S\} \tag{2}$$

with U_b representing the output of the b-th intra-chunk RNN and $Z_b[:, :, i]$ referring to Z_b applied to the chunk dimension S. After passing U_b through

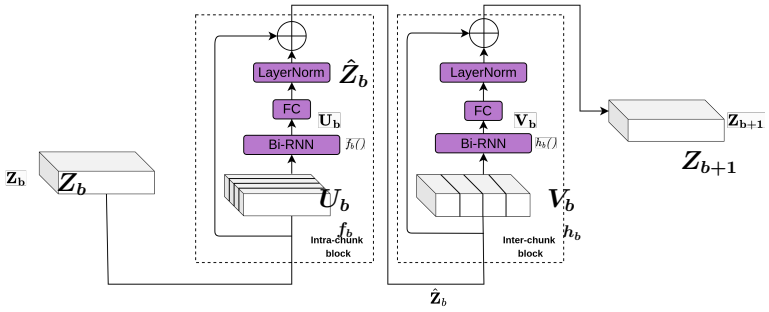

Figure 3: Segmentation process of the DPRNN masking network (figure extracted from [2]).

a linear fully connected (FC) layer (to retrieve the same dimension as that of Z_b) and applying layer normalization (LN), the output $\hat{Z}_b$ is obtained.

$$\hat{Z}_b = Z_b + LN(FC(U_b)) \tag{3}$$

This output is then passed and processed in the same manner on the K dimensions to obtain V_b :

$$V_b = h_b(\hat{Z}_b[:, i, :]) \quad \forall i \in \{1, ..., K\} \tag{4}$$

with $\hat{Z}_b[:, i, :]$ referring to Z_b applied to the chunk length dimension K. Finally, the output of the b-th layer becomes the input of the next one.

$$Z_{b+1} = \hat{Z}_b + LN(FC(V_b)) \tag{5}$$

By alternating the processing either on the chunk size dimension K (intra), or the chunk index dimension S (inter), the model is capable of learning both short and long-term dependencies. After multiple iterations of this process, the 3D tensor is reshaped using overlap-add, to retrieve a signal output.

To further increase the separation capability of ToTaToNet and test its impact on the diarization performance, we replace the current DPRNN with the recent Monaural Speech Separation TransFormer 2 (MossFormer2) [31]. MossFormer2 removes the sequential processing of RNNs to the benefit of self-attention mechanisms that allow for better capturing of the global context. This masking network features two subsequent blocks that are repeated multiple times. The first one, the MossFormer [32], is composed of 4 convolution modules and a main Single-Head Self-Attention (SHSA) layer which serves for both local and global modeling. The convolutional layers receive the normalized and position-embedded features from the TasNet encoder, and their outputs are combined into the SHSA module. A gating mechanism is applied to control the flow of information through the attention layer to capture both local and global context. The second block of MossFormer2 is the Recurrent Module which mainly features a “RNN-free" recurrent network called FSMN (Feedforward Sequential Memory Networks) [33]. The latter has been shown to outperform conventional RNNs and LSTMs in modeling sequential signals, while also modeling longer dependencies.

While typical DPRNN masking networks count approximately 2M to 3M parameters (depending on the TasNet encoder parameters), MossFormer2 consequently increases it to 20M to 50M. Furthermore, due to the attention

mechanisms of the latter, training a ToTaToNet model with MossFormer2 as the masking network takes 2 to 3 times longer than using a traditional DPRNN.

2.4. ASR fine-tuning

While modern ASR models achieve impressive performance on general speech recognition tasks, their effectiveness often deteriorates when confronted with domain-specific scenarios that differ significantly from their training data. Therefore, ASR models are often fine-tuned on in-domain data to make them more accurate in specific use-cases. Domain adaptation through fine-tuning is particularly crucial for multi-party meeting scenarios due to significant divergence from standard ASR training distributions. Meeting environments present distinct technical challenges that necessitate model adaptation, such as speaker overlap, variable signal-to-noise ratios due to speaker-microphone distances, reverberation effects and non-stationary background noises. Multi-party conversations often include non-uniform speaker turns with frequent interruptions, short utterance segments interleaved with overlapping confirmatory responses and extended context dependencies spanning multiple speaker turns. Fine-tuning allows for the adaptation of both acoustic and language models to these domain-specific phenomena.

Fine-tuning ensures model adaptation to the specific characteristics of the target domain data, enabling the system to effectively handle domain-specific phenomena during the decoding phase. In the context of multi-party meeting speech, a primary challenge lies in the handling of overlapping speech: contemporary ASR architectures typically lack the capability to generate multiple parallel token streams corresponding to simultaneous speakers. Instead, these models are commonly trained to process overlapping speech segments sequentially, ordered by utterance onset times. This sequential processing introduces complications in the accurate determination of word timestamps, whether directly from the ASR model or through secondary forced alignment models, as the temporal ordering of individual words becomes non-linear relative to their actual occurrence.

This challenge is specifically associated with single-channel audio input containing multiple overlapping speakers. While speaker separation preprocessing can mitigate this particular issue, it introduces its own set of challenges. When processing speaker-separated streams, the ASR system must contend with reduced contextual information, as each stream contains only

the speech of a single participant within a multi-party conversation. This limitation impacts the model's ability to leverage broader conversational context for improved recognition accuracy. Additionally, the isolation of individual speakers complicates the processing of short backchannel utterances, which often derive meaning from their temporal relationship to other speakers' contributions.

While both single-stream and separated-stream approaches present distinct challenges in multi-party meeting transcription, it is essential to maintain consistency between the preprocessing methodologies applied during training and inference phases. Specifically, if speaker separation is employed during inference, the training data must undergo similar processing. During the training phase, speaker separation can be guided using reference information, such as speaker-attributed word timestamps. In some scenarios, individual speaker-specific microphone recordings are available, providing natural "gold standard" separated streams. In the experiments presented later in the paper, we investigate this preprocessing consistency hypothesis by evaluating ASR models fine-tuned on both single-stream and separated-stream configurations across both testing conditions.

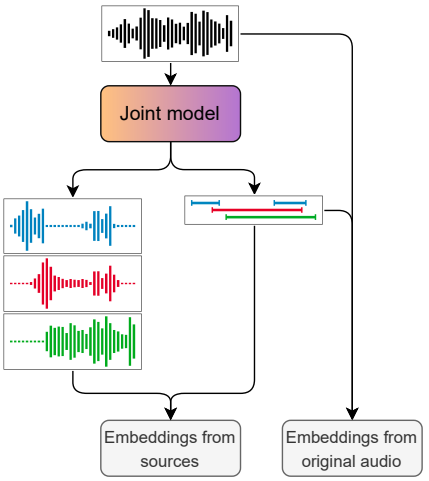

Figure 4: Speaker embedding extraction using either the active frames from separated sources or original audio as input.

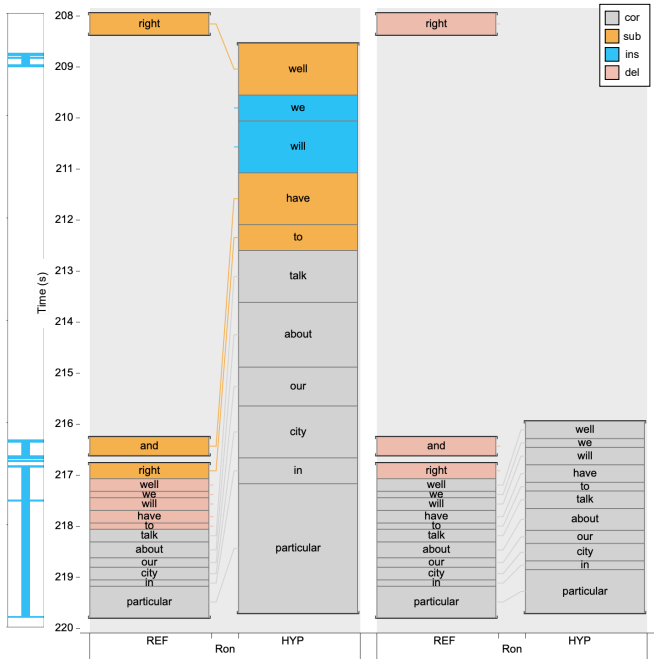

Figure 5: An example of timestamp errors caused by long silences. From left to right: Active speech segments predicted by PixIT, speaker-attributed ASR output's tcpWER alignment visualized before timestamp refinement, and after refinement.

2.5. Separated sources as input to speaker embeddings

In [34], we demonstrated that PixIT diarization performance can improve when speaker embeddings are extracted from the locally separated sources rather than the original audio. In this section, we further investigate this observation. Extracting speaker embeddings from the original audio is clearly disadvantageous in regions with overlapping speech, so it seems natural that separation would be beneficial. However, separated sources may introduce artifacts that could potentially confuse speaker embedding models.

We evaluate and compare the performance of the two embedding extraction approaches (depicted in Figure 4) across multiple datasets for both diarization and SA-ASR tasks. Clustering hyperparameters are optimized

separately on the corresponding development sets for each approach. Results are reported on development sets, and in subsequent sections, we adopt the extraction method that has delivered superior performance for each dataset and task.

2.6. Tackling ASR timestamp errors caused by long silent regions

PixIT's file-level separated sources often contain substantial periods of silence. This issue is particularly pronounced in the NOTSOFAR-1 dataset, where meetings can include up to eight speakers. During our challenge participation, we used faster-whisper[3], a reimplemented version of OpenAI's Whisper decoder that incorporates VAD to remove silent regions before processing the audio. However, this approach introduced a timing issue: sometimes Whisper-assigned word timestamps fall on the incorrect side of a VAD boundary. When this happens, the final timestamps can be shifted by the duration of the removed silent region, leading to large timing errors. These misalignments can cause an increase in tcpWER because they fall outside the acceptable time collar. This issue is illustrated in Figure 5 using MeetEval's tcpWER alignment visualizer.

Faster-whisper includes a heuristic to mitigate this issue. When a word's timestamps span two VAD segments separated by more than two seconds of silence, it is assigned to the segment where its temporal midpoint aligns in the post-VAD timeline. However, this method can still lead to utterances being split between distant VAD segments.

Instead of using Silero VAD[4], as in faster-whisper, we utilize active speech segments from PixIT's diarization output. This approach incurs no additional computational cost and better aligns the VAD train-test domains. To account for artifacts introduced by PixIT's separation, we further refine the heuristic to adjust timestamps only when diarization detects inactivity for more than half of the utterance's duration.

3. Experiments

3.1. Datasets

Our experiments are conducted on three distinct and publicly available datasets, all sourced from meeting recordings captured using a single micro-

[3] `https://github.com/SYSTRAN/faster-whisper`
[4] `https://github.com/snakers4/silero-vad`

phone. The first one, AMI [35], consists of 100 hours of recordings coming from multi- and single-channel microphones, spread across 171 meetings. Part of the dataset consists of scenario-driven meetings, where participants take on predefined roles, while the rest comprises real-life, natural meetings. The second one is AliMeeting [36]. It is a Mandarin corpus comprising approximately 120 hours of natural meeting recordings across 212 sessions.

While AMI contains approximately 15–20% of overlapping speech, AliMeeting presents more challenging scenarios with around 40% overlap. Finally, the NOTSOFAR-1 [21] dataset contains 150 hours of single-channel recordings from 5 different devices and 110 hours of multi-channel recordings from 4 devices, all accounting for a total of 280 meetings. The dataset also includes 1000 hours of tailored synthetic mixtures. Due to computational limitations, we do not utilize this synthetic data for training PixIT in this work. While AMI and AliMeeting both contain audio recordings ranging from 30 to 60 minutes approximately, NOTSOFAR-1 is composed of 6 to 7 minutes long files. Since PixIT is a single-channel method, we focus our experiments on AMI-SDM (Single Distant Microphone version of AMI), AliMeeting *channel 1*, and NOTSOFAR-SC.

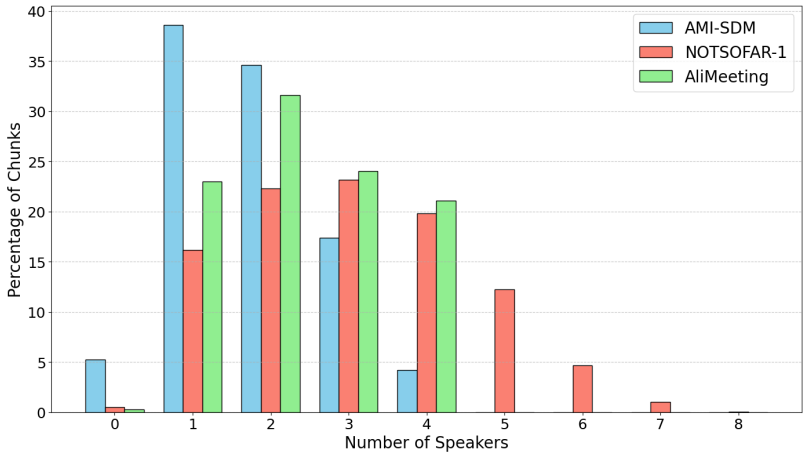

Figure 6: Histogram of the total number of speakers per 5-second chunk across datasets generated on the train sets.

As shown in Figure 6, AMI-SDM contains the smallest amount of speakers

per 5-second chunk, which makes this dataset the least challenging of the three. On the contrary, NOTSOFAR-1 is the most challenging dataset for both diarization and separation, due to a very large amount of active speakers per chunk. Furthermore, the fact that the number of recorded meetings is significantly higher than AMI or AliMeeting adds a lot of diversity in the possible scenarios to solve.

3.2. Evaluation metrics

Our systems were evaluated on the speaker diarization and speaker-attributed transcription task. For the speaker diarization task, we have employed the Diarization Error Rate (DER), which is one of the most used metrics to evaluate and compare systems on this task. The DER is computed as follows:

$$DER = \frac{FA + MISS + SC}{TOTAL}, \tag{6}$$

where FA stands for False Alarm and is equal to the duration of non-speech wrongly classified as speech. MISS is for missed detection and is the opposite case, which means speech is incorrectly classified as non-speech. SC (Speaker Confusion) is the duration correctly classified as speech but assigned to the wrong speaker. Finally, TOTAL is the sum over all speakers of their reference speech duration, so overlapped speech is counted multiple times. We did not apply any collar to compute the DER in the results reported in this article.

To evaluate our systems on the speaker-attributed transcription task, we rely on the concatenated minimum-permutation word error rate (cpWER) [37] and the time-constrained minimum-permutation WER (tcpWER) from MeetEval[38]. Both metrics are an extension of the Word Error Rate, a widely used metric for the transcription task. This one is defined by:

$$WER = \frac{I + S + D}{S + D + C}, \tag{7}$$

with I, S, D, and C respectively the number of inserted, substituted, deleted, and correct words when comparing the hypothesis provided by the system with the reference. WER has been designed for single-speaker ASR systems and thus does not take into account speaker attribution.

The cpWER was introduced to penalize systems on speaker confusion. To compute this metric, the reference and hypothesis segments are first grouped by speaker and then concatenated. Next, the Hungarian algorithm [39] is applied to find the permutation that minimizes the WER, resulting in the

reported cpWER. Unmatched hypothesis and predicted speaker transcripts both count as errors.

Lastly, tcpWER adds a temporal constraint to prevent matching words that are far apart temporally, thereby evaluating the quality of temporal transcript prediction. In our experiments, we use a temporal collar of 5 seconds as is done in the NOTSOFAR-1 Challenge.

When evaluating the transcripts, text normalization is first applied. On AliMeeting and AMI we used Whisper's normalizer. On NOTSOFAR-1 we used the slightly modified Whisper normalizer used in the challenge.

There are two ways of aggregating the above metrics across files. The first is to sum up all the individual components (e.g. FA, MISS, SC, TOTAL for DER) across the files and then calculate the metric. The second is by averaging the metric values across files therefore weighting all files equally. We followed the first approach for all evaluations except NOTSOFAR-1 where we used the second approach to match the challenge evaluation.

3.3. Speaker attribution methods

The standard approach to adding speaker attribution to an off-the-shelf ASR system involves integrating it with a speaker diarization system, as illustrated in Figure 7. Each speech segment produced by the ASR system is assigned to the speaker who is the most active during that segment based on the diarization output. While a CSS system can be used as a preprocessing step to better handle overlapping speech, it requires tailored synthetic training data, as discussed earlier.

The NOTSOFAR-1 challenge baseline includes a CSS-based system trained on a tailored synthetic dataset, which we use for comparison with PixIT. To our knowledge, no publicly available synthetic datasets exist for AMI or AliMeeting. While generating such datasets and training a CSS-based system would strengthen the baseline, this is beyond the scope of our work. Therefore, our baselines for AMI and AliMeeting exclude CSS, relying solely on Whisper for ASR and pyannote.audio 3.1 [40] for diarization.

PixIT, on the other hand, offers a more seamless approach to SA-ASR. It outputs long-form separated sources that are inherently speaker-attributed through alignment with diarization. As a result, SA-ASR with PixIT simply involves passing each long-form source directly into an ASR system. In the following, we will use these two SA-ASR methods for benchmarking PixIT's separation capabilities, since due to the lack of clean reference signals a direct evaluation is not possible on real-world audio.

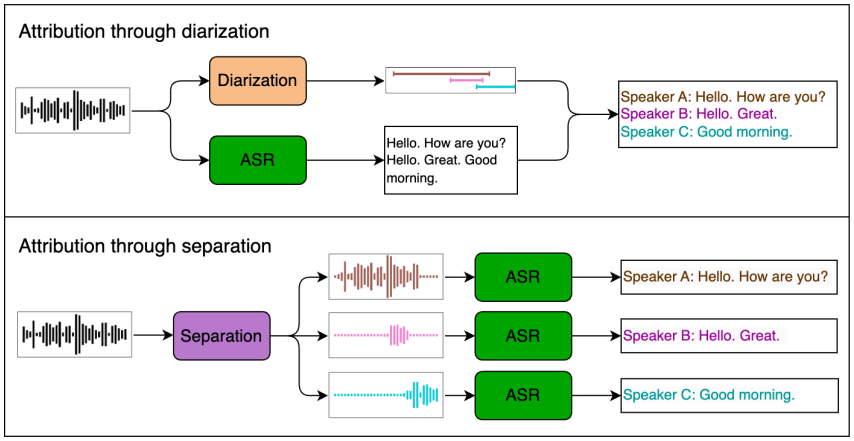

Figure 7: Adding speaker attribution to ASR as performed through either diarization or separation.

3.4. Implementation details

Following the original work, every model has been optimized using the Adam optimizer and employs 3 output masks. The WavLM and W2-BERT components are respectively fine-tuned with a learning rate of 1e-5, while the remainder of the model utilizes a learning rate of 3e-4. Given the size of W2v-BERT, which makes full fine-tuning impractical, we adopt LoRA [41] (Low-Rank Adaptation) with a rank $r = 8$ and scaling factor $\alpha = 32$. Table 1 provides a detailed summary of the different configurations explored.

The training configuration remains consistent across both AMI and AliMeeting systems. Rather than training new systems for NOTSOFAR-1, our focus is on maximizing the potential of our challenge-trained system by enhancing the ASR back-end. All models have been trained using a single A100-80 GB GPU.

Our separation and diarization pipelines utilize speaker embeddings extracted by the ECAPA-TDNN model from Speechbrain [42]. These pipelines require the optimization of three or four key hyperparameters. The first is the segmentation threshold, which determines when a segmentation output is considered active. The other two parameters are critical during the clustering phase: the clustering threshold and the minimum cluster size. Additionally, for the separation pipeline, there is an ASR collar hyperparameter, which

Table 1: Overview of parameter counts, training hyperparameters, and Real Time Factors (RTFs) for the proposed ToTaToNet architectures. RTF is measured by inferring each model on the AMI *development* set.

SSL	Masking network	# Params		Batch size	Learning rate		RTF
		Frozen	Trainable		SSL	Other	
WavLM	DPRNN	0	319M	16	1e-5	3e-4	0.005
WavLM	MossFormer2	0	319M	8	1e-5	3e-4	0.009
W2v-BERT	DPRNN	580M	5M	16	1e-5 (LoRA)	3e-4	0.012
W2v-BERT	MossFormer2	580M	21M	8	1e-5 (LoRA)	3e-4	0.017

specifies the amount of padding applied to the speech segments determined by diarization when leakage removal is performed [19].

We employ Agglomerative Hierarchical Clustering to group similar speakers. In this method, the clustering threshold defines the minimum distance required to separate clusters, while the minimum cluster size parameter dictates the merging of smaller clusters. The optimization process follows a two-step approach. First, we determine the optimal segmentation threshold and ASR collar using Oracle clustering. Next, we optimize the clustering parameters based on this threshold. Optimization is done with respect to DER or cpWER depending on the task.

3.5. Results

3.5.1. Effect of speaker embeddings inputs

The DER and cpWER scores on the development sets of the various datasets are reported in Table 2 for pipelines extracting embeddings from mixtures and sources. We also provide the percentage of overlapped frames for the explored datasets for reference. For the AliMeeting and AMI datasets, we use the original ToTaToNet systems, featuring WavLM and DPRNN. For the NOTSOFAR-1 dataset, we employ our best-performing system submitted to the challenge. To expedite optimization, we use the Whisper `small.en` and `small` models for the ASR system in this evaluation. This approach is also followed in subsequent sections when optimizing SA-ASR systems. In the following, we refer to speaker embeddings extracted from estimated single-speaker intervals of the original mixture as mixture embeddings and those extracted from separated signals as source embeddings.

We observe a general trend: datasets with higher overlap tend to show better performance with source embeddings compared to mixture embed-

dings. Results on AliMeeting exhibit a conflicting pattern where SA-ASR performance improves, but diarization degrades when using separated sources. Conversely, NOTSOFAR-1 shows the opposite trend. Further analysis reveals that merging two clusters to improve speaker diarization can sometimes introduce artifacts into the separated signal that lead to Whisper's hallucinations and significantly worsen the tcpWER score. This highlights a limitation of using SA-ASR for evaluating separation performance.

The fact that extracting embeddings from separated sources can improve performance with an off-the-shelf speaker embedding model is promising. As shown in Section 3.5.4, fine-tuning the ASR system on separated sources, yields a significant improvement to SA-ASR performance – Whisper adapts to the separation artifacts. This suggests promising opportunities for leveraging separated sources in speaker embedding extraction, provided the embedding models are fine-tuned to handle such artifacts effectively.

Table 2: DER (%) and cpWER (%) for different embedding extraction methods across datasets as calculated on the development split.

Dataset	**Overlap (%)**	**Embeddings input**	**DER (%)**				**cpWER (%)**			
			FA	**MD**	**SC**	**total**	**sub**	**del**	**ins**	**total**
AMI	14.6	Mixtures	4.9	6.3	4.8	**16.0**	7.3	20.0	2.3	**29.6**
		Sources	4.9	6.3	8.5	**19.7**	6.7	21.7	2.3	**30.7**
AliMeeting	20.4	Mixtures	4.6	6.6	6.1	**17.4**	16.2	22.2	3.4	**41.8**
		Sources	4.6	6.6	7.1	**18.3**	15.6	22.2	3.2	**41.0**
NOTSOFAR-1	39.4	Mixtures	4.2	9.1	9.4	**22.7**	8.4	22.2	4.2	**34.9**
		Sources	4.2	9.1	8.1	**21.3**	8.2	23.1	4.5	**35.8**

3.5.2. Performance of different ToTaToNet architectures

This section details the separation and diarization performance of the proposed ToTaToNet architectures on the AMI and AliMeeting datasets. Table 3 presents the SA-ASR results on AMI using PixIT-based separation and Whisper *large-v3*. MossFormer2 outperforms DPRNN for both SSL models. Aligning with findings for supervised speech separation [31] reinforcing the generalizability of the joint ToTaToNet framework. An unexpected observation is the underperformance of W2v-BERT SSL features compared to WavLM in all cases. This is analyzed further in Section 3.5.3.

To assess broader applicability, we train the best-performing architecture (WavLM with MossFormer2) on AliMeeting, and the results are displayed in Table 4. These findings underscore the generalizability of the trends observed earlier.

Table 5 provides the diarization results for both datasets. For the new architectures, we only trained the $\lambda = 0.5$ models due to computational constraints. Diarization performance remains consistent across configurations. While more capable masking networks enhance separation performance, this improvement does not appear to directly translate to better diarization. This may be attributed to the relatively simple diarization module currently integrated into ToTaToNet. A comment can be made about the trade-off between performance and architectural size in the masking networks (by looking at the RTFs of Table 1). While MossFormer2 outperforms DPRNN, its results on AMI show only a slight improvement over the baseline while requiring significantly longer training and inference times. This raises concerns about the relevance of such changes for our ToTaToNet systems. On AliMeeting, the performance gain is greater, making the change in the masking network more justified for this particular dataset.

Table 3: tcpWER (%) and cpWER (%) for various ToTaToNet architectures with speaker attribution via diarization or separation evaluated on AMI-SDM dataset using Whisper *large-v3* for ASR.

SSL model	**Masking network**	**Speaker attribution**	**Attribution model**	**cpWER (%)**				**tcpWER (%)**			
				sub	**del**	**ins**	**total**	**sub**	**del**	**ins**	**total**
Not used	Not used	Diarization	pyannote 3.1	7.2	27.8	4.8	**39.7**	6.1	29.5	6.4	**42.0**
WavLM	DPRNN	Diarization	ToTaToNet	7.5	26.0	3.4	**36.9**	6.4	27.8	5.4	**39.5**
		Separation	ToTaToNet	7.0	19.6	2.8	**29.3**	7.3	21.4	4.6	**33.4**
WavLM	MossFormer2	Diarization	ToTaToNet	7.3	26.3	3.3	**36.9**	6.2	28.0	5.0	**39.2**
		Separation	ToTaToNet	6.9	19.4	2.6	**28.9**	7.1	21.3	4.5	**32.9**
W2v-BERT	DPRNN	Diarization	ToTaToNet	7.3	26.5	3.3	**37.1**	6.2	28.2	5.1	**39.4**
		Separation	ToTaToNet	7.3	22.7	2.6	**32.6**	7.7	23.9	4.5	**36.0**
W2v-BERT	MossFormer2	Diarization	ToTaToNet	7.5	26.2	3.2	**36.8**	6.2	28.0	5.0	**39.2**
		Separation	ToTaToNet	7.8	19.6	3.2	**30.6**	7.4	21.8	5.4	**34.7**

3.5.3. Effect of self-supervised models

In this section, we focus on analyzing the performance of both WavLM and W2v-BERT for diarization and separation respectively. Surprisingly, Results in Tables 3 and 5 show that W2v-BERT did not improve over WavLM.

Table 4: tcpCER (%) and cpCER (%) for various ToTaToNet architectures with speaker attribution via diarization or separation evaluated on AliMeeting *channel 1* dataset using Whisper *large-v3* for ASR.

SSL model	Masking network	Speaker attribution	Attribution model	cpCER (%)				tcpCER (%)			
				sub	del	ins	total	sub	del	ins	total
Not used	Not used	Diarization	pyannote 3.1	17.3	38.5	10.0	**65.9**	9.8	46.0	17.5	**73.3**
WavLM	DPRNN	Diarization	ToTaToNet	18.3	37.6	9.2	**65.1**	10.1	45.6	17.2	**72.9**
		Separation	ToTaToNet	10.8	28.6	2.7	**42.1**	13.4	30.9	5.0	**49.4**
WavLM	Mossformer2	Diarization	ToTaToNet	19.1	37.0	8.5	**64.6**	10.3	45.5	17.0	**72.8**
		Separation	ToTaToNet	12.4	25.0	3.3	**40.7**	15.1	28.8	7.1	**51.1**

In fact, the larger SSL model leads to a performance degradation. To further investigate this issue we train from scratch separate segmentation and a separation models.

For the diarization part, we fix the hyperparameters to be the same as in PixIT (same audio chunk size, same batch size, etc). Since we no longer require any TasNet encoder, we directly extract the audio representations from the self-supervised model and pass them through 4 LSTM layers to capture temporal dependencies on the features. Finally, the output of the LSTM is fed into a linear layer to get probabilities of active classes corresponding to each active speaker (mimicking the process in PixIT). The training dataset for the segmentation is AMI, in order to properly compare to PixIT's segmentation performance.

For separation, we use the same masking network (DPRNN) as in PixIT while also fixing the same kernel and stride hyperparameters for the TasNet encoder. Since we want to directly assess the separation capabilities of the tested SSL architectures (without any bias coming from a transcription model), we choose the WSJ0-2Mix dataset in order to evaluate directly the separation gains instead of the cpWER. The latter consists of artificial mixtures created from distinct clean utterances, providing ground truths for the separated sources. Regarding metrics, we evaluate the separation performance of each system by computing the Signal-to-Distortion Ratio (SDR) between the ground truth and each predicted source [**?**]. To get a more relevant evaluation, we also compute the SDR improvement (SDRi) and the Scale-Invariant SDRi (SI-SDRi).

Finally, to maintain comparability to PixIT, W2v-BERT and WavLM are fine-tuned under the same conditions as their respective ToTaToNet models.

Table 5: DER (%) on AMI-SDM and AliMeeting *channel 1* for different ToTaToNet systems trained with PixIT. State-of-the-art as of December 2024 is denoted with 🏆.

SSL model	Masking network	λ	DER (%)			
			FA	MD	SC	total
AMI-SDM systems						
Han et al. [?]						**15.4** 🏆
WavLM	DPRNN	1.0	4.4	7.2	5.5	**17.1**
	DPRNN	0.5	3.9	8.2	5.6	**17.7**
WavLM	MossFormer2	0.5	5.0	8.5	3.9	**17.5**
W2v-BERT	DPRNN	0.5	5.0	8.8	3.9	**17.6**
	MossFormer2	0.5	4.9	8.6	4.2	**17.7**
AliMeeting systems						
Härkönen et al. [43]						**13.2** 🏆
WavLM	DPRNN	1.0	4.7	6.5	8.3	**19.5**
	DPRNN	0.5	5.8	7.3	8.3	**21.4**
WavLM	MossFormer2	0.5	6.8	6.9	7.7	**21.4**

Table 6 shows a clear benefit from using W2v-BERT for audio representations compared to WavLM. For diarization, a relative 10% improvement is observed on the DER metric. Analyzing each sub-component of the DER reveals consistent gains in performance, with a significant boost in the detection of missed speech (MD). For separation, we observe a 14% relative constant improvement in dB from using W2v-BERT instead of WavLM across all different metrics. These results indeed confirm that W2v-BERT is supposed to increase the performance for both tasks, which cannot be said for its implementation in the end-to-end ToTaToNet models.

While [19] demonstrated the benefit of using WavLM to boost both diarization and separation capabilities in PixIT, the experiment conducted on this section highlights a potential bottleneck in the way ToTaToNet models leverage SSL representations. One limitation stems from the way the SSL representations are equally used in both the diarization and separation

Table 6: DERs (%) and gains in dB (higher is better) on various metrics for WavLM (fine-tuned) and W2v-BERT 2.0 (fine-tuned with LoRA) trained disjointly on segmentation and separation models respectively. The segmentation models are trained on AMI-SDM, while the separation models (using DPRNN as the masking network) are trained on WSJ0-2Mix.

SSL model	DER (%)				Gains (dB)		
	FA	MD	SC	total	SDR	SDRi	SI-SDRi
WavLM	4.3	8.8	6.2	**19.2**	16.4	16.2	16.0
W2v-BERT	4.0	7.2	5.8	**17.3**	18.6	18.4	18.2

tasks. To further investigate this aspect, we perform a layer-wise analysis of the used SSL models with regards to the speaker diarization and separation tasks, as well as for the joint task (PixIT). To do so, we freeze the SSL models and apply a weighted average to the 24 "former" layers (such that $\sum \alpha_i = 1$, where α_i represents a scalar associated with i^{th} layer). We train each downstream task until convergence and observe the activations related to each layer.

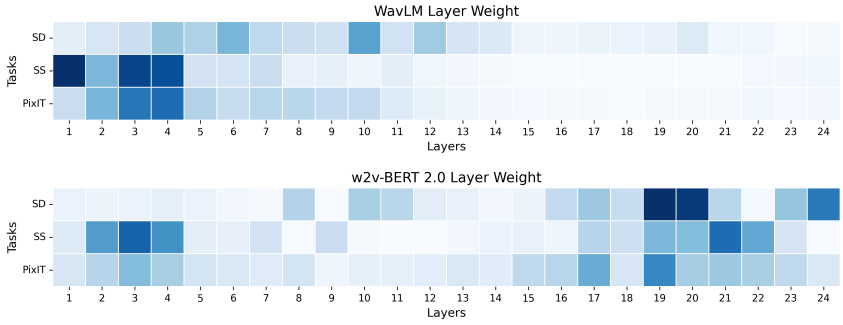

Figure 8: Layer contribution of w2v-BERT 2.0 and WavLM (LARGE version) for Speaker Diarization (SD), Speaker Separation (SS), and the joint task (PixIT)

By looking at the layer contribution of the WavLM model for the downstream tasks of speaker diarization and source separation, we see a clear activation towards the first layers of the architecture (from layer 1 to layer 7

mainly). These layers contribute the most to speaker identity-related tasks. This behavior translates well for PixIT, whose activation is mostly located on the same part as with each task (early layers). On the other hand, when looking at the activations for the W2v-BERT 2.0 model, we see a more heterogenous contribution. For diarization, it is mostly top layers that are active, showing a similar trend as SSL models pretrained on conversational data instead of single-speaker content [44]. For separation, both the early and top layers are active (which is also contradictory to the activations seen for WavLM). The discrepancy observed for the activations related to the two tasks translates to the joint one, as PixIT attempts to leverage both the early and top part of w2v-BERT 2.0. For a joint model like PixIT, which requires a unified set of representations for both diarization and separation, using an SSL model such as W2v-BERT – where layer contributions are concentrated at opposite ends of the architecture – a choice must be made that trades off performance in each of the tasks.

As a result, further investigations and experiments are necessary to explore how features from SSL models can be effectively integrated with ToTaToNet, to ensure an optimal contribution of the SSL model to both tasks.

3.5.4. Fine-tuning ASR

Fine-tuning ASR models on in-domain training data has been shown to yield significant improvements compared to employing universal ASR models. To investigate the impact of fine-tuning using different speaker diarization and separation methods, we created two versions of fine-tuned Whisper *large-v3* models: (1) trained on the original AMI-SDM training data with merged transcripts, and (2) trained on AMI-SDM data, but using separated audio sources and corresponding speaker-attributed transcripts.

The AMI dataset includes word-level timestamps for transcriptions. To convert word-based transcripts into utterances, words attributed to the same speaker were concatenated into sequences, which were then segmented into utterances. Segmentation was based on sentence-ending punctuation marks, utterance length exceeding 10 seconds, or pauses longer than one second between words. For the first model, trained on original SDM data, the segmented utterances from different speakers were merged based on their temporal order. For the second model, trained on separated audio sources, utterances were filtered to exclude those which did not have a corresponding speech segment, as detected by the ToTaToNet model. This filtering primarily affected short backchannels and hesitation sounds which the ToTaToNet

Table 7: cpWER (%) and tcpWER (%) on AMI-SDM test set for various fine-tuned Whisper *large-v3* models, with various attribution methods, including relative changes compared to no fine-tuning.

ASR fine-tuning	Attribution	cpWER (%)				tcpWER (%)				Relat. change (%)	
		sub	del	ins	total	sub	del	ins	total	cpWER	tcpWER
None	Diarization	7.3	26.4	3.2	**36.9**	6.4	27.9	4.9	**39.2**	–	–
On original audio	Diarization	9.9	14.0	7.1	**30.9**	8.2	15.9	9.3	**33.4**	-16.2	-14.8
On separated sources	Diarization	7.8	22.9	2.3	**32.9**	7.1	24.3	3.8	**35.1**	-10.8	-10.5
None	Separation	5.8	21.7	1.7	**29.3**	6.5	22.8	2.8	**32.2**	–	–
On original audio	Separation	14.1	9.5	19.1	**42.8**	11.1	13.1	23.1	**47.3**	+46.1	+47.0
On separated sources	Separation	4.1	16.7	1.8	**22.6**	6.8	14.4	3.7	**24.8**	-22.9	-23.0

often misses.

Both fine-tuned Whisper *large-v3* models were trained with an identical hyperparameter configuration, which had not been optimized specifically for the AMI dataset but was based on prior experiments with other datasets. The training data was divided into 30-second segments, and the models were trained for three epochs using the AdamW optimizer. Each epoch used an effective batch size of 64 segments, a peak learning rate of 10^{-5}, 50 warm-up steps, and a linear learning rate decay schedule. Stochastic weight averaging [45] was applied during the final epoch, using a constant learning rate of 10^{-6}.

Table 7 presents cpWER and tcpWER results for the AMI-SDM test set, comparing various Whisper models and speaker separation configurations. When the original single-channel SDM audio is used as input, both fine-tuned models demonstrate improvements over the non-fine-tuned model. Notably, the model fine-tuned on the original single-channel audio, which aligns with the test-time configuration, achieves superior performance. In contrast, when Whisper models are applied to each separated source independently, the differences between models become more pronounced. Surprisingly, the Whisper model fine-tuned on multi-speaker single-channel audio results in a noticeable decline in ASR accuracy. Conversely, the model fine-tuned on separated sources delivers a substantial improvement. This indicates that when employing PixIT to separate multi-speaker ASR test data into speaker-specific channels, it is crucial to fine-tune the ASR model on separated audio that reflects the test-time configuration.

3.5.5. Improving on our NOTSOFAR-1 Challenge submission

Table 8 presents cpWER and tcpWER metrics for our NOTSOFAR-1 Challenge systems evaluated on the eval-small dataset. It includes the relative error increases from the collar, changes after applying the timestamp fix heuristic detailed in Section 2.6, and comparisons to the baseline system post-heuristic. The effect of the timestamp fix heuristic is analyzed in detail in Section 3.5.6.

The ToTaToNet checkpoint used here is always the same as in our submission for the challenge. Unlike in preceding sections, results are calculated by averaging error rates across files, as done in the challenge evaluation.

Fine-tuning the *large-v3* model on the original single-channel audio results in an even bigger increase in both cpWER and tcpWER than for AMI. One explanation for this might be that the number of file-level separated sources output by PixIT is larger for NOTSOFAR-1. Based on eyeing the output and the huge insertion error rate, the issues for this system seems to be caused by frequent hallucination. We experimented with using Whisper's hallucination detection functionality but this yielded limited improvement.

Conversely, fine-tuning on separated sources ("large-v3, ft. on sep. sources") notably improves the tcpWER to 33.7%, achieving a 20% relative reduction compared to the baseline, thereby again demonstrating the effectiveness of aligning the fine-tuning process with the test-time configuration.

Notably, we are able to slightly improve on the NOTSOFAR-1 baseline using identical downstream ASR (Whisper *large-v2*). Thus PixIT is a promising alternative to CSS even in cases where carefully constructed domain-matched synthetic datasets are available.

3.5.6. Effect of the time-stamp fix heuristic

The effect of our time-stamp fix heuristic is detailed in Table 9. The application of the heuristic generally mitigates error increases, with the most pronounced improvements observed in the separated sources fine-tuned model. While further gains could likely be achieved through more refined post-processing heuristics, these results demonstrate that despite PixIT exhibiting a higher relative proportion of timestamp errors, such errors can largely be corrected without retraining by applying lightweight post-processing techniques.

Table 8: Detailed performance of systems on the eval-small split. This table includes cpWER with its components, tcpWER with its components, and the relative tcpWER change with respect to the baseline system.

System	cpWER (%)				tcpWER (%)				ΔtcpWER (%) (relative)
	sub	del	ins	total	sub	del	ins	total	
NOTSOFAR-1 baseline	11.3	22.0	7.4	40.7	10.0	23.3	8.8	42.1	0.0
Our NOTSOFAR-1 submission	10.7	14.2	9.8	34.7	10.3	17.6	13.2	41.1	-2.4
large-v2	7.7	25.4	3.7	36.8	7.4	27.9	6.3	41.7	-1.0
large-v3	7.1	24.9	3.7	35.6	7.2	27.5	6.3	40.9	-2.9
large-v3, ft. on single channel	21.8	14.0	45.0	80.8	14.2	21.4	52.4	88.1	+109.3
large-v3, ft. on sep. sources	8.0	16.3	6.0	30.3	7.2	18.3	8.1	33.7	-20.0

Table 9: Effect of time-stamp fix heuristic on tcpWER metric. This table presents the total cpWER, tcpWER before and after applying the fix, along with the relative collar error proportion before the fix and the relative change to collar errors from fixing.

System	cpWER (%)	tcpWER (%)		Rel. collar err. proportion (%)	Rel. change to collar err. from fix (%)
		before fix	after fix		
large-v3	35.6	41.6	40.9	13.2	-11.7
large-v3, ft. on sep. sources	30.3	34.8	33.7	12.9	-24.4

4. Conclusion

This work evaluates PixIT, a joint training approach for supervised speaker diarization and unsupervised speech separation, and builds on our systems for the NOTSOFAR-1 Challenge to demonstrate its effectiveness. We examine alternative choices for the joint ToTaToNet in terms of masking networks and self-supervised learning (SSL) features. We show that MossFormer2 improves separation performance over DPRNN like in the supervised separation case. Although W2v-BERT has been shown to improve the two tasks independently, it underperforms for joint training. This might be due to mismatched layers needed for separation and diarization, highlighting the need for more careful integration of these features into the joint architecture.

We show that fine-tuning ASR systems on PixIT-separated sources significantly boosts downstream SA-ASR performance. Notably, the gains are bigger than for a standard diarization-based SA-ASR system when ASR is fine-tuned on original mixtures. We also demonstrate that time-stamp errors produced by PixIT-based SA-ASR can be effectively mitigated with

lightweight post-processing.

PixIT-separated sources prove useful for speaker embedding extraction in the diarization pipeline, achieving results comparable to embeddings from original mixtures. Considering how well it works for SA-ASR, fine-tuning speaker embeddings on separated sources appears to be a promising area for improvement.

We apply the aforementioned techniques to our NOTSOFAR-1 Challenge submission to achieve a 20% tcpWER improvement over the CSS-based baseline by 20% without using any of the provided domain-specific synthetic data that the CSS system was trained on. Even with the same ASR model as the baseline, PixIT outperforms it, showing that in addition to being much easier to train, it is able to rival supervised separation approaches on real-world mixtures.

5. Acknowledgments

This work was granted access to the TalTech supercomputing resources and the HPC resources of IDRIS under the allocations AD011014044R1 and AD011014274R1 made by GENCI, and supported by the Agence de l'Innovation Défense under the grant number 2022 65 0079. All experiments also benefited from the support of the French National Research Agency through the ANR-20-CE23-0012-01 (MIM) grant and the Estonian Centre of Excellence in AI.

References

[1] Y. Luo, N. Mesgarani, Conv-TasNet: Surpassing ideal time-frequency magnitude masking for speech separation, IEEE ACM Trans. Audio Speech Lang. Process. 27 (8) (2019) 1256–1266.

[2] Y. Luo, Z. Chen, T. Yoshioka, Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation, in: 2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, 2020, pp. 46–50.

[3] C. Xu, W. Rao, X. Xiao, E. S. Chng, H. Li, Single channel speech separation with constrained utterance level permutation invariant training using grid LSTM, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, 2018, pp. 6–10.

[4] L. Drude, D. Hasenklever, R. Haeb-Umbach, Unsupervised training of a deep clustering model for multichannel blind source separation, in: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, 2019, pp. 695–699.

[5] N. Ito, S. Araki, T. Nakatani, Complex angular central gaussian mixture model for directional statistics in mask-based microphone array signal processing, in: 24th European Signal Processing Conference, EUSIPCO 2016, 2016, pp. 1153–1157.

[6] S. Wisdom, E. Tzinis, H. Erdogan, R. J. Weiss, K. W. Wilson, J. R. Hershey, Unsupervised sound separation using mixture invariant training, in: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, 2020.

[7] A. Sivaraman, S. Wisdom, H. Erdogan, J. R. Hershey, Adapting speech separation to real-world meetings using mixture invariant training, in: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, 2022, pp. 686–690.

[8] Z. Chen, T. Yoshioka, L. Lu, T. Zhou, Z. Meng, Y. Luo, J. Wu, X. Xiao, J. Li, Continuous speech separation: Dataset and analysis, in: 2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, 2020, pp. 7284–7288.

[9] M. Rouvier, P. Bousquet, B. Favre, Speaker diarization through speaker embeddings, in: 23rd European Signal Processing Conference, EUSIPCO 2015, 2015, pp. 2082–2086.

[10] F. Landini, J. Profant, M. Díez, L. Burget, Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: Theory, implementation and analysis on standard tasks, Comput. Speech Lang. 71 (2022) 101254.

[11] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, S. Watanabe, End-to-end neural speaker diarization with permutation-free objectives, in: 20th Annual Conference of the International Speech Communication Association, Interspeech 2019, 2019, pp. 4300–4304.

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

[13] K. Kinoshita, M. Delcroix, N. Tawara, Advances in integration of end-to-end neural and clustering-based diarization for real conversational speech, in: 22nd Annual Conference of the International Speech Communication Association, Interspeech 2021, 2021, pp. 3565–3569.

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

[15] Y. Li, X. Zheng, P. C. Woodland, Self-supervised learning-based source separation for meeting data, in: IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, 2023, pp. 1–5.

[16] T. von Neumann, K. Kinoshita, M. Delcroix, S. Araki, T. Nakatani, R. Haeb-Umbach, All-neural online source separation, counting, and diarization for meeting analysis, in: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, 2019, pp. 91–95.

[17] S. Maiti, Y. Ueda, S. Watanabe, C. Zhang, M. Yu, S. Zhang, Y. Xu, EEND-SS: joint end-to-end neural speaker diarization and speech separation for flexible number of speakers, in: IEEE Spoken Language Technology Workshop, SLT 2022, 2022, pp. 480–487.

[18] Y. Bando, T. Nakamura, S. Watanabe, Neural Blind Source Separation and Diarization for Distant Speech Recognition, in: 25th Annual Conference of the International Speech Communication Association, Interspeech 2024, 2024, pp. 722–726.

[19] J. Kalda, C. Pagés, R. Marxer, T. Alumäe, H. Bredin, PixIT: Joint training of speaker diarization and speech separation from real-world multi-speaker recordings, in: Odyssey 2024: The Speaker and Language Recognition Workshop, 2024, pp. 115–122.

[20] J. Kalda, T. Alumäe, S. Baroudi, M. Lebourdais, H. Bredin, R. Marxer, ToTaTo system descriptions for the NOTSOFAR1 challenge, in: 8th International Workshop on Speech Processing in Everyday Environments (CHiME 2024), 2024, pp. 23–25.

[21] A. Vinnikov, A. Ivry, A. Hurvitz, I. Abramovski, S. Koubi, I. Gurvich, S. Pe'er, X. Xiao, B. M. Elizalde, N. Kanda, X. Wang, S. Shaer, S. Yagev, Y. Asher, S. Sivasankaran, Y. Gong, M. Tang, H. Wang, E. Krupka, NOTSOFAR-1 challenge: New datasets, baseline, and tasks for distant meeting transcription, CoRR abs/2401.08887 (2024). arXiv:2401.08887.

[22] Y. Luo, N. Mesgarani, TaSNet: Time-domain audio separation network for real-time, single-channel speech separation, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, 2018, pp. 696–700.

[23] J. L. Roux, S. Wisdom, H. Erdogan, J. R. Hershey, SDR - half-baked or well done?, in: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, 2019, pp. 626–630.

[24] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, F. Wei, WavLM: Large-scale self-supervised pre-training for full stack speech processing, IEEE J. Sel. Top. Signal Process. 16 (6) (2022) 1505–1518.

[25] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, 2019, pp. 4171–4186.

[26] W. Hsu, B. Bolte, Y. H. Tsai, K. Lakhotia, R. Salakhutdinov, A. Mohamed, HuBERT: Self-supervised speech representation learning by masked prediction of hidden units, IEEE ACM Trans. Audio Speech Lang. Process. 29 (2021) 3451–3460.

[27] Y. Chung, Y. Zhang, W. Han, C. Chiu, J. Qin, R. Pang, Y. Wu, w2v-BERT: Combining contrastive learning and masked language modeling for self-supervised speech pre-training, in: IEEE Automatic Speech

Recognition and Understanding Workshop, ASRU 2021, 2021, pp. 244–250.

[28] Seamless Communication, L. Barrault, Y.-A. Chung, M. C. Meglioli, D. Dale, N. Dong, M. Duppenthaler, P.-A. Duquenne, B. Ellis, H. El-Sahar, J. Haaheim, J. Hoffman, M.-J. Hwang, H. Inaguma, C. Klaiber, I. Kulikov, P. Li, D. Licht, J. Maillard, R. Mavlyutov, A. Rakotoarison, K. R. Sadagopan, A. Ramakrishnan, T. Tran, G. Wenzek, Y. Yang, E. Ye, I. Evtimov, P. Fernandez, C. Gao, P. Hansanti, E. Kalbassi, A. Kallet, A. Kozhevnikov, G. M. Gonzalez, R. S. Roman, C. Touret, C. Wong, C. Wood, B. Yu, P. Andrews, C. Balioglu, P.-J. Chen, M. R. Costa-jussà, M. Elbayad, H. Gong, F. Guzm'an, K. Heffernan, S. Jain, J. T. Kao, A. Lee, X. Ma, A. Mourachko, B. Peloquin, J. Pino, S. Popuri, C. Ropers, S. Saleem, H. Schwenk, A. Y. Sun, P. Tomasello, C. Wang, J. Wang, S. Wang, M. Williamson, Seamless: Multilingual expressive and streaming speech translation, ArXiv abs/2312.05187 (2023).

[29] A. Baevski, Y. Zhou, A. Mohamed, M. Auli, wav2vec 2.0: A framework for self-supervised learning of speech representations, in: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, 2020.

[30] A. Gulati, J. Qin, C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, R. Pang, Conformer: Convolution-augmented transformer for speech recognition, in: 21st Annual Conference of the International Speech Communication Association, Interspeech 2020, 2020, pp. 5036–5040.

[31] S. Zhao, Y. Ma, C. Ni, C. Zhang, H. Wang, T. H. Nguyen, K. Zhou, J. Q. Yip, D. Ng, B. Ma, MossFormer2: Combining transformer and RNN-Free recurrent network for enhanced time-domain monaural speech separation, in: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2024, 2024, pp. 10356–10360.

[32] S. Zhao, B. Ma, MossFormer: Pushing the performance limit of monaural speech separation using gated single-head transformer with convolution-augmented joint self-attentions, in: IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, 2023, pp. 1–5.

[33] S. Zhang, C. Liu, H. Jiang, S. Wei, L. Dai, Y. Hu, Feedforward sequential memory networks: A new structure to learn long-term dependency, CoRR abs/1512.08301 (2015). arXiv:1512.08301.

[34] J. Kalda, T. Alumäe, M. Lebourdais, H. Bredin, S. Baroudi, R. Marxer, TalTech-IRIT-LIS speaker and language diarization systems for DISPLACE 2024, in: Interspeech 2024, 2024, pp. 1635–1639.

[35] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. M. Post, D. Reidsma, P. Wellner, The AMI meeting corpus: A pre-announcement, in: Machine Learning for Multimodal Interaction, Second International Workshop, MLMI 2005, Vol. 3869 of Lecture Notes in Computer Science, 2005, pp. 28–39.

[36] F. Yu, S. Zhang, Y. Fu, L. Xie, S. Zheng, Z. Du, W. Huang, P. Guo, Z. Yan, B. Ma, X. Xu, H. Bu, M2Met: The icassp 2022 multi-channel multi-party meeting transcription challenge, in: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, 2022, pp. 6167–6171.

[37] S. Watanabe, M. Mandel, J. Barker, E. Vincent, A. Arora, X. Chang, S. Khudanpur, V. Manohar, D. Povey, D. Raj, D. Snyder, A. S. Subramanian, J. Trmal, B. B. Yair, C. Boeddeker, Z. Ni, Y. Fujita, S. Horiguchi, N. Kanda, T. Yoshioka, N. Ryant, Chime-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings, in: 6th International Workshop on Speech Processing in Everyday Environments (CHiME 2020), 2020, pp. 1–7.

[38] T. von Neumann, C. B. Boeddeker, M. Delcroix, R. Haeb-Umbach, Meeteval: A toolkit for computation of word error rates for meeting transcription systems, in: 7th International Workshop on Speech Processing in Everyday Environments (CHiME 2023), 2023, pp. 27–32.

[39] H. W. Kuhn, The hungarian method for the assignment problem, Naval research logistics quarterly 2 (1-2) (1955) 83–97.

[40] A. Plaquet, H. Bredin, Powerset multi-class cross entropy loss for neural speaker diarization, in: 24th Annual Conference of the International

Speech Communication Association, Interspeech 2023, 2023, pp. 3222–3226.

[41] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-rank adaptation of large language models, in: The Tenth International Conference on Learning Representations, ICLR 2022, 2022.

[42] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, Y. Bengio, SpeechBrain: A general-purpose speech toolkit, arXiv:2106.04624 (2021). arXiv:2106.04624.

[43] M. Härkönen, S. J. Broughton, L. Samarakoon, EEND-M2F: Masked-attention mask transformers for speaker diarization, in: 25th Annual Conference of the International Speech Communication Association, Interspeech 2024, 2024, pp. 37–41.

[44] S. Baroudi, T. Pellegrini, H. Bredin, Specializing self-supervised speech representations for speaker segmentation, in: 25th Annual Conference of the International Speech Communication Association, Interspeech 2024, 2024, pp. 3769–3773.

[45] P. Izmailov, D. Podoprikhin, T. Garipov, D. P. Vetrov, A. G. Wilson, Averaging weights leads to wider optima and better generalization, in: Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, 2018, pp. 876–885.

Appendix 5

V

Joonas Kalda, Clément Pagés, Tanel Alumäe, and Hervé Bredin. Diarization-Guided multi-speaker embeddings. In *26th Annual Conference of the International Speech Communication Association, Interspeech 2025*, pages 5233–5237, 2025

Diarization-Guided Multi-Speaker Embeddings

Joonas Kalda[1], *Clément Pagés*[2], *Tanel Alumäe*[1], *Hervé Bredin*[2]

[1]Tallinn University of Technology, Estonia
[2]IRIT, Université de Toulouse, CNRS, Toulouse, France

joonas.kalda@taltech.ee

Abstract

Reliable speaker embeddings are critical for multi-speaker speech processing tasks. Traditionally models are trained on single-speaker utterances and suffer from domain mismatch when applied in multi-speaker contexts. Recently proposed guided speaker embeddings (GSE) were shown to improve this by training on synthetic multi-speaker mixtures guided by oracle speaker activity labels. Additionally modeling all speakers present in a chunk is desirable but the performance of such methods has been sub-par up to now. We build on GSE by modeling multiple speakers together and using diarization features for guiding. We also propose a new validation metric for embeddings in multi-speaker context and demonstrate its effectiveness. Results on multiple speaker diarization datasets demonstrate that we improve on speed and performance while reducing the embedding model size.

Index Terms: speaker diarization, speaker verification

1. Introduction

High-quality speaker embeddings are essential for multi-speaker speech processing tasks. In speaker diarization, EEND-vector clustering (EEND-VC) systems rely on speaker embeddings derived from local segmentation outputs to cluster and stitch these local segments [1]. Similarly, in multi-speaker automatic speech recognition (ASR), transducer-based systems produce segment-wise transcriptions with timestamp estimates, which are subsequently attributed to individual speakers using speaker embeddings [2]. For voice conversion of long-form audio, it is desirable that speaker embeddings are modelled consistently [3]. In all the above use cases, it would be beneficial for all the speakers to be modeled concurrently.

Previous studies have explored joint training of ASR and segmentation models with multi-speaker embeddings [2, 4, 5], but these approaches have underperformed compared to standalone embedding systems [6]. This discrepancy is likely due to the difference in data quality: speaker verification datasets, which are more easily annotated, tend to be larger and more diverse than those used for ASR and speaker diarization [7]. These datasets contain only single-speaker utterances, leading to a domain mismatch when applied in multi-speaker scenarios. To address this, guided speaker embeddings (GSE) were recently introduced for multi-speaker environments [8]. GSE is trained on synthetic multi-speaker mixtures derived from a speaker verification dataset, with oracle activity labels guiding the process. Activity labels for both target and interference speakers are used as additional inputs to the embedding encoder and for masking attention scores. However, these systems still produce embeddings for only one speaker at a time.

This work proposes extending GSE by modeling all speakers present in a chunk at once. Additionally, since the practical deployment of a GSE system relies on a speaker segmentation model, we also propose using its output as a guide for training instead of oracle labels. Features from a segmentation model can offer more detailed guidance. For example, areas of high confidence indicate it is easier to extract speaker-specific information there.

The main contributions of this work are as follows:

- Proposing a diarization-guided training method for multi-speaker embedding systems.
- Introducing a modified attention module to allow for multi-speaker modeling in existing speaker embedding models.
- Proposing a new validation metric optimized for speaker embeddings in a multi-speaker context.
- Providing a thorough evaluation of the multi-speaker embeddings on multiple speaker diarization and verification datasets.
- Open-sourcing the code for the above[1].

2. Method

Figure 1 illustrates our joint architecture, which combines a local speaker segmentation model with a speaker embedding model using a shared feature extractor. We opt for an SSL-based features extractor, namely WavLM, since it demonstrates good performance in both speaker diarization and speaker verification tasks [9]. It is also the choice for state-of-the-art for speaker diarization as of writing [10, 11]. We use the same LSTM-based segmentation probing head as in [11]. For the embedding module, we use an ECAPA-TDNN, which has been shown to perform better than smaller probing heads [12, 13].

Given the frame-level features extracted from an audio chunk $\mathbf{x} \in \mathbb{R}^{T \times F}$ and assuming a maximum of $K_{\max}$ speakers, the segmentation module extracts powerset features $\mathbf{p} \in \mathbb{R}^{T \times K_{\text{ps}}}$, where K_{ps} is the number of powerset classes. These are binarized and converted into a multi-label format $\mathbf{a} \in \{0, 1\}^{T \times K_{\max}}$ [14]. The powerset features are concatenated with $\mathbf{x}$ to form combined features of dimension $F + K_{\text{ps}}$, which are fed into the embedding encoder. Only the input channel dimension of the encoder is modified in our approach.

The encoder output $\mathbf{h} \in \mathbb{R}^{T \times D}$ is reshaped to introduce a speaker dimension, resulting in $\mathbf{h}' \in \mathbb{R}^{K_{\max} \times T \times (D/K_{\max})}$. The attention module remains unchanged from the original ECAPA-TDNN, except that all the channel dimensions are scaled down by a factor of $K_{\max}$, except for the bottleneck attention dimension, which is kept at 128. The batch size after the encoder is effectively increased $K_{\max}$ times, with the speakers being processed in parallel.

[1]https://github.com/joonaskalda/multi-speaker-embeddings

Similarly to GSE, we apply silent masking for each predicted speaker but use binarized predicted speaker activations instead of oracle labels. The embedding dimension for the predicted multi-speaker embeddings $\{\mathbf{e}_1, \ldots, \mathbf{e}_{K_{\max}}\} \in \mathbb{R}^{192}$ is kept unchanged from the original ECAPA-TDNN.

Note that this approach would require slight modifications if the encoder output channel dimension D is not divisible by $K_{\max}$ by e.g. adding an adaptation layer. In the above we also assumed, for simplicity, that the embedding encoder leaves the temporal resolution unchanged, as is the case for ECAPA-TDNN, used in our experiments. If that is not the case, the speaker activation masks should be interpolated to match the embedding output temporal resolution.

To train the multi-speaker embedding model we use synthetic VoxCeleb mixtures as in [8]. We use the standard ArcFace loss [15] but only compute it for an embedding if the segmentation model correctly predicts the corresponding speaker's activation for at least one second.

In our experiments, we use a segmentation probing head trained using powerset loss [14], but this is not a requirement. The only new components that need to be trained in our method are in the speaker embedding branch. For the segmentation branch and feature extractor, any off-the-shelf model can be utilized, and no specific adaptation is needed. Our proposed training method generalizes naturally to any local segmentation and speaker embedding architecture, with no requirement for a shared feature extractor.

To summarize, our method builds on top of GSE by

- Changing the attention module to extract multi-speaker embeddings instead of single-speaker embeddings.
- Utilizing detailed information from the segmentation module.

2.1. Validation metric

The standard validation metric for speaker embedding models is the equal error rate (EER), computed on single-speaker utterance trials. However, this does not reflect performance in multi-speaker scenarios. We argue that speaker diarization performance is a more appropriate metric, as the clustering stage directly depends on embedding quality. We therefore propose using diarization performance for both evaluation and validation. Validation is challenging; normally, it would require hyperparameter optimization after each epoch, which is a resource-intensive process.

To address this, we propose a simplified pipeline for validation (Figure 2). In each validation batch, all audio chunks are sampled from the same file using a sliding window, ensuring that the first chunk starts at the beginning of the audio and the last one stops at the file's end. The step size S between consecutive chunks is chosen so that all chunks are evenly spaced i.e. $S = \frac{D-T}{B-1}$, where D is the file duration, B is the batch size, and T is the chunk length. Chunks overlap if and only if $BT > D$.

The batch is then fed to the model, which returns segmentation predictions and speaker embeddings for each chunk. These embeddings are then clustered using the K-means algorithm, where the number of clusters is fixed based on the oracle number of speakers in the file used to create the batch. For efficiency, we assume the number of speakers is known, which eliminates the need for tuning clustering parameters. Finally, segments are assigned to speakers as in a standard diarization pipeline, and a diarization error rate (DER) over the validation files is calculated based on the pipeline output and the corresponding cropped reference.

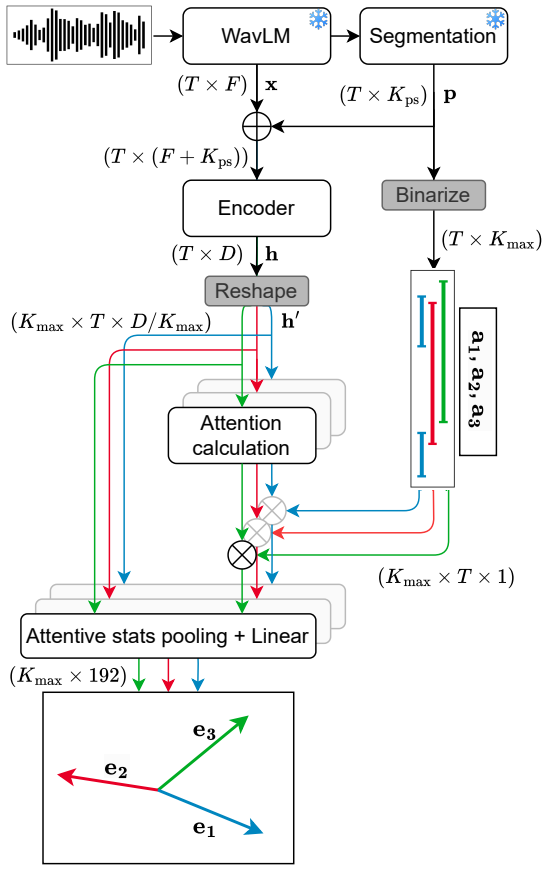

Figure 1: *Proposed joint architecture for a maximum of $K_{\max} = 3$ speakers per audio chunk. We opt for a segmentation branch trained using a powerset loss, but this is not a requirement.*

3. Experiments

3.1. Datasets

The feature extractor and diarization branch are trained on a composite dataset consisting of AMI-SDM [16], AliMeeting (first channel) [17], AISHELL-4 (first channel) [18], MSD-WILD [19], RAMC [20], and EGO4D [21]. Since EGO4D does not include an evaluation set, we use it only for training and validation.

The speaker embedding systems are trained on either VoxCeleb 1 and 2 utterances [7, 22] or synthetic mixtures generated from these datasets.

3.2. Data simulation

For training speaker embedding systems in multi-speaker contexts, we use 10-second synthetic VoxCeleb mixtures, following the approach in GSE. However, we modify the simulation method to better reflect real-world multi-speaker scenarios. Specifically, we allow arbitrary speaker order and permit delays

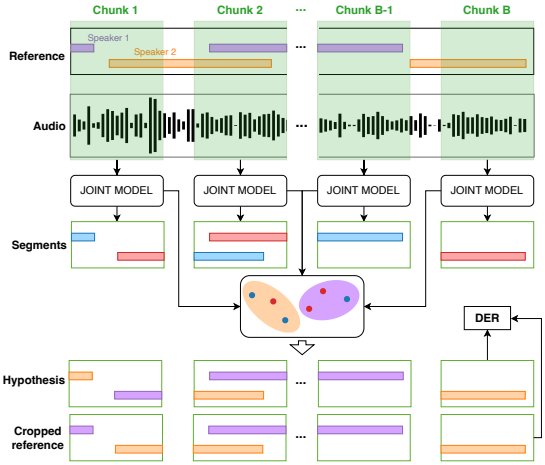

Figure 2: *Proposed validation pipeline with a batch size of B.*

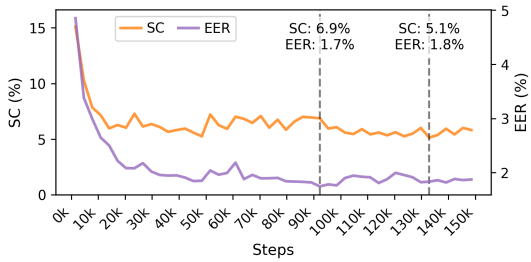

Figure 3: *Validation EER and speaker confusion (SC) (%) scores as a function of step count. The optimal checkpoints based on either metric are highlighted with a dotted line.*

of up to 0.5 seconds after the preceding utterance ends to introduce natural silent regions. Utterance lengths are sampled from an exponential distribution with $\lambda = 0.2$, truncated to the range [1,10] seconds. Additionally, we apply room background noise to the mixtures using data from [23] and simulated room impulse responses from [24].

3.3. Implementation details

Segmentation model. Training chunks are 10 seconds long, with a maximum of $K_{\text{max}} = 3$ speakers per chunk. We train a standard diarization system using powerset loss, assuming that no more than two speakers are active at a time, resulting in $K_{\text{ps}} = 7$ powerset classes, assuming no more than two concurrent speakers. Our segmentation module follows the architecture from [11].

We use a WavLM Base+ model as the shared feature extractor, fine-tuned together with the segmentation module as in [10]. The learning rates are set to 10^{-5} for WavLM and 3×10^{-4} for the segmentation module, with a batch size of 32. The embedding and segmentation models use separate learnable weighted sums of the WavLM layers.

Speaker embedding model. Our speaker embedding extractor is an ECAPA-TDNN model with 1024 channels. As a baseline, we train an unguided single-embedding system on 3-second utterances with a batch size of 512.

For all other speaker embedding systems, we adopt the training strategy from [8]. We employ the Adam optimizer with a cyclical learning rate schedule over three cycles, using a batch size of 128 mixtures. This results in an effective batch size of 384 for ArcFace loss computation. Each cycle consists of 50k steps, beginning with a 1k-step warm-up phase, followed by cosine annealing decay. The learning rate starts at a peak of 10^{-3} and decays by a factor of 0.75 at the start of each new cycle.

Validation metrics. For our proposed DER-based validation metric, we randomly sample 10 files from each dataset's validation set, yielding a total of 58 batches [2]. The baseline validation metric is the equal error rate (EER), calculated on the widely used VoxCeleb test set 1-O, containing 37611 test trials based on single-speaker utterances [7].

Speaker diarization inference. For speaker diarization inference, we use the pyannote 3.1 pipeline [25] with the same configuration as [8]. After selecting the optimal checkpoint based on the validation metric, we optimize the speaker diarization clustering hyperparameters for each system using Optuna [26]. Hyperparameter tuning is performed on the compound validation set using the multivariate Tree-structured Parzen Estimator for 100 iterations.

Evaluation. Direct evaluation of speaker embeddings in a multi-talker context would require a multi-talker real-world verification dataset, which currently does not exist. Previous work has used synthetic mixtures based on VoxCeleb to assess multi-speaker performance [8], but these do not accurately capture real-world conversational dynamics [27]. Because of this, we assess embedding quality indirectly via diarization pipeline performance. With a fixed local segmentation model, false alarm and missed detection rates are constant. Consequently, we only report speaker confusion rates, determined by clustering and directly reflecting embedding quality. Scores on the VoxCeleb test set 1-O are also reported for reference.

4. Results

A comparison of multi-speaker and single-speaker embeddings, along with different guiding methods, is presented in Table 1. All guided systems outperform the standard unguided system in diarization but underperform in EER, consistent with the findings of [8]. Switching from oracle-guided single-embedding to multi-embedding leads to a performance drop, which is expected since the encoder must now model all participating speakers rather than a single target speaker, while the model size is slightly reduced (due to scaling down the channel dimension in the attention module). However, this degradation is mitigated by replacing oracle-guided training with diarization-guided training.

Multi-speaker models are also more compact, as the channel dimension is scaled down by K_{max} after the encoder. The oracle clustering system serves as an upper bound, assuming perfect speaker clustering, with non-zero speaker confusion scores arising only from intra-chunk segmentation errors. Validation using the proposed simplified diarization pipeline demonstrates clear improvements in most cases, with comparable results for the GSE system, where the selected checkpoints had very similar performance.

Figure 3 shows the EER and speaker confusion scores as a

[2] AliMeeting validation set only has 8 files

	Training guide	Params	Validation Metric	EER (%)	AISHELL4	AMI-SDM	AliMeeting	MSDWILD	RAMC	Macro-avg.
Oracle clustering	-	-	-	-	1.3	1.9	2.3	2.8	3.5	2.5
Single-embedding	Unguided	24.3M	EER	1.1	4.7	8.3	10.2	12.6	8.1	8.4
			DER	1.4	5.1	8.0	8.3	12.2	7.9	8.1
Single-embedding [8]	Oracle	24.3M	EER	1.8	2.9	3.8	3.6	12.1	7.3	5.9
			DER	1.9	2.7	3.7	3.6	11.8	7.2	5.9
Multi-embedding	Oracle	22.5M	EER	1.6	3.0	6.2	4.2	13.4	6.7	6.5
			DER	2.2	3.2	5.6	4.4	11.2	7.1	6.2
Multi-embedding	Diarization	22.5M	EER	1.7	3.2	3.7	5.0	11.4	7.3	6.2
			DER	1.8	2.5	3.8	3.5	11.4	7.2	5.7

Table 1: *Comparison of single-speaker (single-embedding) and multi-speaker (multi-embedding) embedding systems with different guiding mechanisms and validation metrics. We report EER on VoxCeleb 1-O and speaker confusion rates on diarization datasets, as well as the macro-average (Macro-avg) for the latter. Speaker confusion using oracle clustering is included as a topline reference.*

Dataset	MSE	ResNet34	SOTA
AISHELL-4	12.0	12.4	10.6 [11]
AMI-SDM	15.7	16.5	15.4 [10]
AliMeeting	15.9	17.4	13.2 [28]
MSDWILD	22.9	21.6	19.6 [11]
RAMC	11.8	11.1	11.1 [28]
Macro-average	15.0	15.1	13.4

Table 2: *Comparison of DERs for multi-speaker embeddings (MSE), and ResNet34 embeddings across datasets. State-of-the-art (SOTA) DERs are provided for reference.*

Dataset	Ovr. (%)	Spk. #	RTF Imp. (%)
AISHELL-4	5.0	2.0	39
AMI-SDM	14.6	2.2	43
AliMeeting	20.4	2.8	53
MSDWILD	12.4	2.0	40
RAMC	9.4	1.8	36
Macro-average	12.0	2.1	42

Table 3: *Comparison of overlapping speech percentage (Ovr.), average speaker count per chunk (Spk. #), and relative RTF improvements (RTF Imp.) across datasets.*

function of step count for the diarization-guided multi-speaker embedding system. The two curves display low correlation after initial fast convergence, further highlighting that the standard VoxCeleb 1-O EER is not optimal in multi-speaker applications.

In Table 2, we compare our diarization-guided multi-embedding system to a state-of-the-art ResNet-based speaker embedding model [29] from pyannote 3.1 based on DER, keeping the segmentation model the same. State-of-the-art DER scores are also provided for reference. Although the ResNet system employs a more sophisticated training strategy, including speed augmentation and large-margin fine-tuning, our system displays competitive results across the board, with significantly better results on the higher-overlap datasets AMI-SDM and AliMeeting.

Table 3 explores the real-time factor (RTF) of multi-speaker embeddings compared to single-speaker embeddings, which requires encoding each speaker in a chunk separately. We first measure the total time for inference using a diarization pipeline with oracle speaker clustering, where no speaker embeddings are calculated. Then we measure the increase in RTF from performing clustering using either system. Comparing the results for the two systems gives us the relative decrease in RTF. We also report both the percentage of frames containing overlapped speech and the average number of speakers in a 10-second chunk sampled from the dataset. The latter directly reflects the number of separate forward passes required by the single-speaker embedding encoder, in contrast to the single pass needed for our multi-speaker approach. Even for the relatively low-overlap scenarios represented by AISHELL-4 and RAMC, the multi-embedding system achieves an RTF relative improvement of at least 36%.

4.1. Future work

Training of our diarization-guided multi-speaker embeddings relies on synthetically generated mixtures, which, while useful, fail to capture the complexity of real-world conversation dynamics [27]. Training or fine-tuning the embeddings directly on real-world data should help performance, although a comparatively small number of speakers in real-world conversational datasets poses a challenge here.

We keep the speaker embedding encoder unchanged from the single-speaker case, but since it now has to model multiple speakers, the architecture should be optimized for this.

5. Conclusion

This work introduces a novel diarization-guided training method for multi-speaker embeddings. We extend guided speaker embeddings by modeling speakers concurrently using diarization-based guidance. We also introduced a novel clustering-based validation metric for training embeddings in a multi-speaker context, which we showed to be more effective than standard speaker verification EER based on single-speaker utterances. Keeping the embedding encoder unchanged, we compared the effects of both modifications on multiple speaker diarization datasets. We showed that while switching to modeling multiple speakers concurrently degrades performance, this performance deficit is offset by switching to diarization-based guiding, which contains more information and matches testing conditions. We end up with a speaker embedding system that is smaller, more accurate, and considerably faster than comparable systems trained using previous methods.

6. Acknowledgment

This work was granted access to the HPC resources of IDRIS under the allocation AD011014274R1 made by GENCI, as well as the TalTech supercomputing resources. This work was supported by the Estonian Centre of Excellence in Artificial Intelligence (EXAI).

7. References

[1] 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, pp. 3565–3569.

[2] N. Kanda, J. Wu, Y. Wu, X. Xiao, Z. Meng, X. Wang, Y. Gaur, Z. Chen, J. Li, and T. Yoshioka, "Streaming multi-talker ASR with token-level serialized output training," in *Proc. Interspeech*, 2022, pp. 3774–3778.

[3] E. Casanova, J. Weber, C. D. Shulby, A. C. Junior, E. Gölge, and M. A. Ponti, "Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone," in *Proc. ICML*, 2022, pp. 2709–2720.

[4] N. Kanda, J. Wu, Y. Wu, X. Xiao, Z. Meng, X. Wang, Y. Gaur, Z. Chen, J. Li, and T. Yoshioka, "Streaming speaker-attributed ASR with token-level speaker embeddings," in *Proc. Interspeech*, 2022, pp. 521–525.

[5] K. Kinoshita, M. Delcroix, and N. Tawara, "Integrating end-to-end neural and clustering-based diarization: Getting the best of both worlds," in *Proc. ICASSP*, 2021, pp. 7198–7202.

[6] N. Tawara, M. Delcroix, A. Ando, and A. Ogawa, "Ntt speaker diarization system for chime-7: Multi-domain, multi-microphone end-to-end and vector clustering diarization," in *Proc. ICASSP*, 2024, pp. 11 281–11 285.

[7] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "VoxCeleb: Large-scale speaker verification in the wild," *Computer Speech & Language*, vol. 60, p. 101027, 2020.

[8] S. Horiguchi, T. Moriya, A. Ando, T. Ashihara, H. Sato, N. Tawara, and M. Delcroix, "Guided speaker embedding," in *Proc. ICASSP*, 2025, pp. 1–5.

[9] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[10] J. Han, F. Landini, J. Rohdin, A. Silnova, M. Diez, and L. Burget, "Leveraging self-supervised learning for speaker diarization," in *Proc. ICASSP*, 2025, pp. 1–5.

[11] S. Baroudi, T. Pellegrini, and H. Bredin, "Specializing self-supervised speech representations for speaker segmentation," in *Proc. Interspeech*, 2024, pp. 3769–3773.

[12] S. Zaiem, Y. Kemiche, T. Parcollet, S. Essid, and M. Ravanelli, "Speech self-supervised representations benchmarking: a case for larger probing heads," *Computer Speech & Language*, vol. 89, p. 101695, 2025.

[13] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," in *Proc. Interspeech*, 2020, pp. 3830–3834.

[14] A. Plaquet and H. Bredin, "Powerset multi-class cross entropy loss for neural speaker diarization," in *Proc. Interspeech*, 2023, pp. 1736–1740.

[15] J. Deng, J. Guo, J. Yang, N. Xue, I. Kotsia, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," *TPAMI*, vol. 44, no. 10, pp. 5962–5979, 2022.

[16] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, and M. Kronenthal, "The AMI Meetings Corpus," in *Proc. Symposium on Annotating and Measuring Meeting Behavior*, 2005.

[17] 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*, 2022, pp. 6167–6171.

[18] 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, pp. 1736–1740.

[19] T. Liu, S. Fan, X. Xiang, H. Song, S. Lin, J. Sun, T. Han, S. Chen, B. Yao, S. Liu, Y. Wu, Y. Qian, and K. Yu, "MSDWild: Multi-modal Speaker Diarization Dataset in the Wild," in *Proc. Interspeech*, 2022, pp. 1476–1480.

[20] 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, pp. 1736–1740.

[21] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, M. Martin, T. Nagarajan, I. Radosavovic, S. K. Ramakrishnan, F. Ryan, J. Sharma, M. Wray, M. Xu, E. Z. Xu, C. Zhao, S. Bansal, D. Batra, V. Cartillier, S. Crane, T. Do, M. Doulaty, A. Erapalli, C. Feichtenhofer, A. Fragomeni, Q. Fu, C. Fuegen, A. Gebreselasie, C. Gonzalez, J. Hillis, X. Huang, Y. Huang, W. Jia, W. Khoo, J. Kolar, S. Kottur, A. Kumar, F. Landini, C. Li, Y. Li, Z. Li, K. Mangalam, R. Modhugu, J. Munro, T. Murrell, T. Nishiyasu, W. Price, P. R. Puentes, M. Ramazanova, L. Sari, K. Somasundaram, A. Southerland, Y. Sugano, R. Tao, M. Vo, Y. Wang, X. Wu, T. Yagi, Y. Zhu, P. Arbelaez, D. Crandall, D. Damen, G. M. Farinella, B. Ghanem, V. K. Ithapu, C. V. Jawahar, H. Joo, K. Kitani, H. Li, R. Newcombe, A. Oliva, H. S. Park, J. M. Rehg, Y. Sato, J. Shi, M. Z. Shou, A. Torralba, L. Torresani, M. Yan, and J. Malik, "Ego4D: Around the World in 3,000 Hours of Egocentric Video," in *Proc. CVPR*, 2022, p. 18995–19012.

[22] A. Nagrani, J. S. Chung, J. Huh, A. Brown, E. Coto, W. Xie, M. McLaren, D. A. Reynolds, and A. Zisserman, "VoxSRC 2020: The second VoxCeleb speaker recognition challenge," arXiv:2012.06867, 2020.

[23] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[24] 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 *Proc. ICASSP*, 2017, pp. 5220–5224.

[25] H. Bredin, R. Yin, J. M. Coria, G. Gelly, P. Korshunov, M. Lavechin, D. Fustes, H. Titeux, W. Bouaziz, and M.-P. Gill, "`pyannote.audio`: neural building blocks for speaker diarization," in *Proc. ICASSP*, 2020, pp. 7124–7128.

[26] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. KDD*, 2019, p. 2623–2631.

[27] 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 *Proc. ICASSP*, 2023, pp. 1–5.

[28] M. Härkönen, S. J. Broughton, and L. Samarakoon, "EEND-M2F: Masked-attention mask transformers for speaker diarization," in *Proc. Interspeech*, 2024, pp. 37–41.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

Curriculum vitae

Personal data

Name	Joonas Kalda
Date and place of birth	2 October 1996 Tallinn, Estonia
Nationality	Estonian

Contact information

Address	Tallinn University of Technology, School of Information Technologies, Department of Software Science, Ehitajate tee 5, 19086 Tallinn, Estonia
Phone	+372 620 2165
E-mail	joonas.kalda@taltech.ee

Education

2020–...	Tallinn University of Technology, School of Information Technologies, Department of Software Science, PhD
2018–2019	University of Cambridge, mathematics, MMath
2015–2018	University of Cambridge, mathematics, BA

Language competence

Estonian	native
English	fluent

Professional employment

2020–	Tallinn University of Technology, School of Information Technologies, Department of Software Science, Junior Research Fellow
2023–2023	Institut de Recherche en Informatique de Toulouse, Visiting Research Fellow

Honours and awards

- 2024, Best Student Paper Award at Speaker Odyssey
- 2024, Winner of DISPLACE Challenge at Interspeech

Defended theses

- 2019, "Machine Learning for Classification of Astronomical Time Series", MSc, supervisor Professor Kaisey Mandel, University of Cambridge

Papers

1. Joonas Kalda and Tanel Alumäe. Collar-aware training for streaming speaker change detection in broadcast speech. In *Odyssey 2022: The Speaker and Language Recognition Workshop*, pages 141–147, 2022
2. Joonas Kalda, Clément Pagés, Ricard Marxer, Tanel Alumäe, and Hervé Bredin. PixIT: Joint training of speaker diarization and speech separation from real-world multi-speaker recordings. In *Odyssey 2024: The Speaker and Language Recognition Workshop*, pages 115–122, 2024
3. Joonas Kalda, Tanel Alumäe, Martin Lebourdais, Hervé Bredin, Séverin Baroudi, and Ricard Marxer. TalTech-IRIT-LIS speaker and language diarization systems for DISPLACE 2024. In *25th Annual Conference of the International Speech Communication Association, Interspeech 2024*, pages 1635–1639, 2024
4. Joonas Kalda, Séverin Baroudi, Martin Lebourdais, Clément Pagés, Ricard Marxer, Tanel Alumäe, and Hervé Bredin. Design choices for PixIT-based speaker-attributed ASR: Team ToTaTo at the NOTSOFAR-1 challenge. *Computer Speech & Language*, page 101824, 2026
5. Joonas Kalda, Clément Pagés, Tanel Alumäe, and Hervé Bredin. Diarization-Guided multi-speaker embeddings. In *26th Annual Conference of the International Speech Communication Association, Interspeech 2025*, pages 5233–5237, 2025

Elulookirjeldus

Isikuandmed

Nimi	Joonas Kalda
Sünniaeg ja -koht	2. oktoober 1996, Tallinn, Eesti
Kodakondsus	Eesti

Kontaktandmed

Aadress	Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, Tarkvarateaduse instituut, Ehitajate tee 5, 19086 Tallinn, Eesti
Telefon	+372 620 2165
E-post	joonas.kalda@taltech.ee

Haridus

2020–...	Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, Tarkvarateaduse instituut, doktorantuur
2018–2019	University of Cambridge, matemaatika, magistriõpe (MMath)
2015–2018	University of Cambridge, matemaatika, bakalaureuseõpe (BA)

Keelteoskus

eesti keel	emakeel
inglise keel	kõrgtase

Teenistuskäik

2020–	Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, Tarkvarateaduse instituut, nooremteadur
2023–2023	Institut de Recherche en Informatique de Toulouse, külalisteadur

Autasud

- 2024, Parima üliõpilasartikli auhind, Speaker Odyssey
- 2024, DISPLACE Challenge võitja, Interspeech

Kaitstud lõputööd

- 2019, "Machine Learning for Classification of Astronomical Time Series", magistritöö, juhendaja Professor Kaisey Mandel, University of Cambridge

Artiklid

1. Joonas Kalda and Tanel Alumäe. Collar-aware training for streaming speaker change detection in broadcast speech. In *Odyssey 2022: The Speaker and Language Recognition Workshop*, pages 141–147, 2022
2. Joonas Kalda, Clément Pagés, Ricard Marxer, Tanel Alumäe, and Hervé Bredin. PixIT: Joint training of speaker diarization and speech separation from real-world multi-speaker recordings. In *Odyssey 2024: The Speaker and Language Recognition Workshop*, pages 115–122, 2024
3. Joonas Kalda, Tanel Alumäe, Martin Lebourdais, Hervé Bredin, Séverin Baroudi, and Ricard Marxer. TalTech-IRIT-LIS speaker and language diarization systems for DISPLACE 2024. In *25th Annual Conference of the International Speech Communication Association, Interspeech 2024*, pages 1635–1639, 2024
4. Joonas Kalda, Séverin Baroudi, Martin Lebourdais, Clément Pagés, Ricard Marxer, Tanel Alumäe, and Hervé Bredin. Design choices for PixIT-based speaker-attributed ASR: Team ToTaTo at the NOTSOFAR-1 challenge. *Computer Speech & Language*, page 101824, 2026
5. Joonas Kalda, Clément Pagés, Tanel Alumäe, and Hervé Bredin. Diarization-Guided multi-speaker embeddings. In *26th Annual Conference of the International Speech Communication Association, Interspeech 2025*, pages 5233–5237, 2025

ISSN 2585-6901 (PDF)
ISBN 978-9916-80-464-3 (PDF)