TSVAD+ - A Transformer based approach for Speaker Diarization

A thesis
Submitted towards the partial fulfillment of
BS-MS dual degree programme
by

Ansh Kapadia
20201268

Date: 15/03/2025

under the guidance of

PROF. CHNG ENG SIONG

Nanyang Technological University, Singapore

from July 2024 to Mar 2025

Indian Institute of Science Education and Research Pune

Certificate

This is to certify that this dissertation entitled "TSVAD+ - A Transformer based approach for Speaker Diarization" submitted towards the partial fulfillment of the BS-MS degree at the Indian Institute of Science Education and Research, Pune represents original research carried out by Ansh Jay Kapadia at the Nanyang Technological University, under the supervision of Chng Eng Siong during academic year July 2024 to March 2025.

Supervisor:
CHNG ENG SIONG
PROFESSOR
NTU, SINGAPORE

ANSH JAY KAPADIA
20201268
BS-MS
IISER PUNE

Expert:
AMIT APTE
PROFESSOR
IISER PUNE

DATE: 27/03/2025

Declaration

I, hereby declare that the matter embodied in the report titled "TSVAD+ - A Transformer based approach for Speaker Diarization" is the results of the investigations carried out by me at the Nanyang Technological University, Singapore from the period 15-07-2024 to 31-03-2025 under the supervision of Chng Eng Siong and the same has not been submitted elsewhere for any other degree.

Supervisor:
CHNG ENG SIONG
PROFESSOR
NTU, SINGAPORE

ANSH JAY KAPADIA
20201268
BS-MS
IISER PUNE

DATE: 27/03/2025

Acknowledgements

I would like to express my deepest gratitude to my supervisor, **Prof. Chng Eng Siong**, for his invaluable guidance, insightful discussions, and continuous support throughout the course of this research. I am also immensely grateful to my mentor, **Tran The Anh**, for his patience, technical insights, and hands-on guidance throughout this project. His extensive knowledge and willingness to help at every stage, from conceptualization to experimentation, have greatly enriched my understanding of speaker diarization and deep learning techniques. I would like to thank **Prof. Amit Apte** for his insightful feedback during my mid-year presentation. His valuable suggestions and critical insights helped refine my approach and strengthen the direction of this research. I would also like to acknowledge my lab member, **Adnan Azmat**, whose support in debugging, implementing methodologies, and sharing insightful discussions have significantly contributed to the progress of this work.

I am profoundly grateful to my family—my mother, **Trupti Kapadia**, and my father, **Jay Kapadia**—for their unwavering support and love. Their sacrifices, guidance, and belief in my potential have been the bedrock of my journey. I am equally indebted to my closest friends, **Shreya Mehra** and **Naren S. Narayanan**, whose constant support has been a source of strength. Despite the distance, Naren has always been there to offer perspective, motivation, and a much-needed dose of humor. **Shreya**, in particular, has been an irreplaceable presence in this journey. Her unwavering faith in me, insightful advice, and tireless support have helped me navigate the most challenging moments. Through every setback and success, she has been my strongest pillar, reminding me of my capabilities when I doubted myself and pushing me forward with her encouragement. I would like to extend my gratitude to my friends **Irene Biju** and **Urbi Paul** for their kindness and support in helping me adjust to a new environment in Singapore. I am also incredibly grateful to the members of the Speech and Language Lab, whose guidance, collaboration, and willingness to help have made this journey smoother. Additionally, I would like to thank **NUS** and **NTU** for hosting me and providing an enriching academic environment that enabled me to pursue my research. Lastly, I thank **IISER Pune** for providing this opportunity to work in an esteemed research institution.

To everyone—family, friends, mentors, and colleagues—who have played a role in this journey, your support has been invaluable, and I will always hold deep gratitude for it.

Contributions

Contributor name	**Contributor role**
Tran The Anh, Chng Eng Siong	Conceptualization Ideas
Ansh Jay Kapadia, Adnan Azmat, Tran The Anh	Methodology
Ansh Jay Kapadia, Adnan Azmat, Tran The Anh	Software
Ansh Jay Kapadia, Adnan Azmat, Tran The Anh	Validation
Ansh Jay Kapadia, Adnan Azmat, Tran The Anh	Formal analysis
Ansh Jay Kapadia, Adnan Azmat, Tran The Anh	Investigation
-	Resources
Ansh Jay Kapadia, Adnan Azmat, Tran The Anh	Data Curation
Ansh Jay Kapadia, Adnan Azmat, Tran The Anh	Writing - original draft preparation
Ansh Jay Kapadia, Tran The Anh, Chng Eng Siong	Writing - review and editing
Ansh Jay Kapadia	Visualization
Tran The Anh, Chng Eng Siong	Supervision
Tran The Anh, Chng Eng Siong	Project administration
Chng Eng Siong	Funding acquisition

This contributor syntax is based on the Journal of Cell Science CRediT Taxonomy[1]

[1]https://journals.biologists.com/jcs/pages/author-contributions

List of Figures

1.1 Input-Output of Speaker Diarization System [4]. 5

2.1 Clustering-based diarization pipeline [24]. 9
2.2 Example of a 2-speaker EEND model architecture [13]. 11
2.3 TS-VAD Model Architecture [25]. 13

3.1 TS-VAD+ Model Architecture [25]. 17
3.2 TS-VAD+ Model dataflow [1]. 19
3.3 Memory Aware Attention Architecture 22
3.4 mm-TS-VAD+ Model Architecture [25]. 23

6.1 Illustration of Scaled dot product attention and Multi-head attention by Vaswani et al.[35] 42

List of Tables

2.1 Roadmap of Speaker Diarization Systems and their performance on DIHARD III eval set (track 2) 15

4.1 DER score comparison with different speaker embedding models 27
4.2 Domain wise DER score comparison of different TS-VAD+ models 28
4.3 Domain wise DER score comparison of TS-VAD+ models pretrained on different simulated data 29
4.4 Domain wise DER score comparison using different clustering results . . 30
4.5 DER score breakdown with and without VAD postprocessing 31
4.6 % DER score comparison of TS-VAD+ vs mm-TS-VAD+ 31
4.7 % DER score comparison of TS-VAD+ vs mm-TS-VAD+ 33
4.8 DER score comparison of different models 33

Abstract

Speaker diarization, the task of determining "who spoke when" in an audio recording, is a critical component in applications such as meeting transcription, voice assistant technologies, and conversational analysis. Traditional clustering-based diarization methods struggle with overlapping speech, while end-to-end neural diarization (EEND) systems often lack robustness across diverse acoustic conditions. This thesis presents TS-VAD+, an enhanced transformer-based speaker diarization model that builds upon the TS-VAD framework by incorporating state-of-the-art speaker embeddings (ECAPA-TDNN), a WavLM-based speech encoder, and memory-aware attention mechanisms. These improvements aim to address key limitations in handling multi-speaker and overlapping speech scenarios.

We evaluate the TS-VAD+ model on the DIHARD III dataset, demonstrating its effectiveness through systematic experiments. Pretraining on wideband simulated data (16 kHz) significantly improved domain adaptation, outperforming narrowband-pretrained models. Further refinements, including VBx clustering, voice activity detection (VAD) postprocessing, and data augmentation. While the memory module TS-VAD+ (mm-TS-VAD+) showed promising results in leveraging external speaker embeddings, its performance gains were limited by the size of the fine-tuning dataset.

Overall, TS-VAD+ demonstrates competitive performance in speaker diarization, particularly in high-overlap conditions. Future work could explore self-supervised speaker embeddings, dynamic memory mechanisms, and large-scale augmentation strategies to further enhance diarization accuracy and generalization across diverse domains.

Contents

1 Introduction **4**
- 1.1 What is Speaker Diarization . . . 4
- 1.2 Motivation . . . 4
- 1.3 Contributions . . . 6
  - 1.3.1 Improved Speaker Embeddings . . . 6
  - 1.3.2 Pretraining on Wideband Simulated Data . . . 6
  - 1.3.3 Voice Activity Detection (VAD) Postprocessing . . . 6
  - 1.3.4 Enhanced Clustering for Speaker Detection . . . 6
  - 1.3.5 Memory Module for Speaker Embeddings (mm-TS-VAD+) . . . 7
  - 1.3.6 Data Augmentation for Robustness . . . 7
  - 1.3.7 Performance Improvement on DIHARD III . . . 7

2 Literature Review **8**
- 2.1 Diarization Error Rate . . . 8
- 2.2 Clustering-Based Diarization System . . . 9
- 2.3 End-to-End Neural Diarization (EEND) . . . 11
- 2.4 Target-Speaker Voice Activity Detection (TS-VAD) . . . 13
- 2.5 Overview of Diarization systems throughout the years . . . 15

3 NTU's TS-VAD+ Model **16**
- 3.1 Idea . . . 16
- 3.2 Architecture: . . . 17
- 3.3 Memory Module TS-VAD+ . . . 19
  - 3.3.1 Memory Module Construction . . . 20
  - 3.3.2 Memory-Aware Attention Mechanism . . . 20
  - 3.3.3 Integration into the TS-VAD+ model . . . 21

4 Results and Discussion **24**
- 4.1 Datasets . . . 24
  - 4.1.1 DIHARDIII dataset . . . 24
  - 4.1.2 Narrowband Simulated Data . . . 24
  - 4.1.3 Wideband Simulated Data . . . 25
- 4.2 Training Procedure . . . 25

4.3 Experimental results . . . 26
4.3.1 Evaluation of Speaker Embedding Models for TS-VAD+ . . . 26
4.3.2 Comparison of TS-VAD+ models . . . 27
4.3.3 Wideband simulated data . . . 27
4.3.4 Improvements due to better Clustering . . . 28
4.3.5 Postprocessing using Voice Activity Detection (VAD) . . . 30
4.3.6 Memory Module TS-VAD+ (mm-TS-VAD+) . . . 31
4.3.7 Data Augmentation of DIHARDIII development set . . . 32
4.3.8 Summary of results . . . 33

5 Conclusion and Future Work **34**

References **36**

6 Appendix **40**
A.1 Attention Mechanisms . . . 40
A.1.1 Scaled Dot-Product Attention Mechanism . . . 40
A.1.2 Advantages of Self-Attention . . . 41
A.1.3 Multi-Head Attention . . . 41
A.2 Transformer Encoder Layer . . . 43
A.2.1 Multi-Head Self-Attention . . . 43
A.2.2 Position-Wise Feedforward Network . . . 43
A.2.3 Residual Connections and Layer Normalization . . . 43
A.2.4 Applications . . . 44
A.3 Permutation Invariant Training (PIT) . . . 44
A.3.1 Mathematical Formulation . . . 44
A.3.2 Computational Considerations . . . 45
A.4 Rich Transcription Time Marked File Format . . . 45
A.4.1 RTTM File Structure . . . 45
A.4.2 Example RTTM Entry . . . 46
A.4.3 Usage in Speaker Diarization . . . 46

Chapter 1

Introduction

Since the Stone Age, speech has been one of the foundations for human communication. We share stories, preserve memories, and build relationships through spoken language. However, speech goes beyond the words we use; it carries layers of meaning, emotions, and contextual cues that reveal much about the speaker and the environment in which the conversation occurs. This is the fascinating world of Speaker Diarization, which enables us not only to unravel who spoke when but also to understand other aspects of the conversation in a more nuanced manner.

1.1 What is Speaker Diarization

Speaker diarization is derived from "diarize", which means making notes or keeping an event in a diary. It involves automatically detecting the number of speakers in an audio stream and segmenting it based on their speaking turns, often without prior knowledge of the speakers or the total number of participants. Figure 1.1 illustrates the input and final output of a speaker diarization process.
Speaker diarization systems perform diarization by examining various acoustic features from the audio signal. These features may include Mel-Frequency Cepstral Coefficients (MFCCs), which reflect the spectral properties of speech, as well as pitch and energy contours that reveal aspects of the speaker's voice quality. By analyzing these features across different segments of the recording, speaker diarization systems can identify transitions between speakers and cluster similar segments. In addition, advanced methods such as deep learning are increasingly utilized to enhance accuracy and manage challenging situations, such as overlapping speech or background noise.

1.2 Motivation

Speaker diarization, the process of determining "who spoke when" in an audio recording, is a fundamental problem in speech processing with wide-ranging applications, including

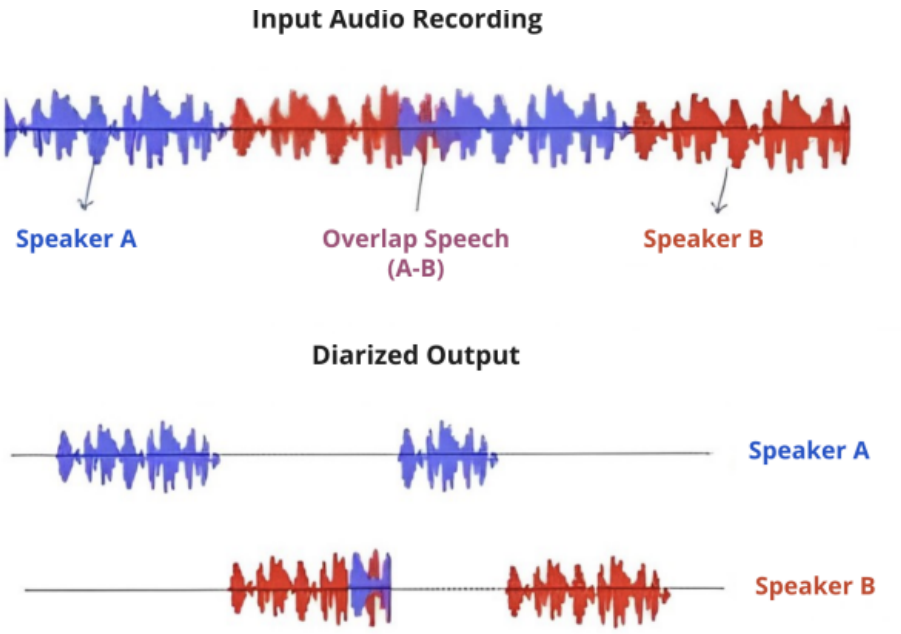

Figure 1.1: Input-Output of Speaker Diarization System [4].

meeting transcription, voice assistants, and conversational analytics. As human interactions increasingly shift to digital platforms, the ability to accurately segment and attribute speech in multi-speaker environments has become a critical challenge.

Traditional clustering-based diarization systems, which rely on handcrafted features and statistical models, often struggle with overlapping speech and varying acoustic conditions. End-to-end neural diarization (EEND) approaches attempt to overcome these limitations by learning to assign speaker labels directly from the input audio. However, these models require extensive training on diverse datasets to generalize effectively across different domains and speaker configurations. Furthermore, they typically do not store explicit speaker embeddings, making cross-session speaker tracking difficult.

Target-Speaker Voice Activity Detection (TS-VAD) has emerged as a promising approach that leverages pre-enrolled speaker embeddings to improve diarization performance. However, existing TS-VAD models face limitations, such as reliance on fixed speaker counts and suboptimal feature extraction techniques. This thesis aims to enhance the TS-VAD framework by integrating state-of-the-art transformer-based architectures, advanced speaker embedding models, and memory-aware attention mechanisms to improve speaker attribution, particularly in high-overlap conditions.

By addressing these challenges, the proposed TS-VAD+ model seeks to advance the state-of-the-art in speaker diarization, improving robustness, scalability, and generaliza-

tion to real-world audio scenarios. The contributions of this work have the potential to significantly impact applications in automated transcription, forensic analysis, and human-computer interaction, where accurate speaker segmentation is essential.

1.3 Contributions

This thesis presents TS-VAD+, an enhanced transformer-based speaker diarization model designed to address key limitations in existing diarization methods, particularly in handling overlapping speech and improving speaker tracking. The primary contributions of this work are as follows:

1.3.1 Improved Speaker Embeddings

Traditional TS-VAD models rely on i-vector embeddings, which are limited in capturing speaker characteristics under challenging conditions. This work replaces i-vectors with ECAPA-TDNN, a state-of-the-art speaker embedding model that provides more robust and discriminative representations. A comparative evaluation of different speaker embedding architectures—including ERes2Net, CAM++, and ECAPA-TDNN—demonstrates that ECAPA-TDNN consistently achieves the lowest Diarization Error Rate (DER), enhancing speaker attribution accuracy.

1.3.2 Pretraining on Wideband Simulated Data

Most diarization models are trained on narrowband (8 kHz) speech data, which often leads to domain adaptation issues when applied to real-world datasets recorded at 16 kHz. This thesis introduces a new wideband (16 kHz) simulated dataset based on the LibriSpeech corpus, featuring dynamic speaker mixing and realistic conversational structures. Experimental results show that pretraining on wideband data significantly improves domain adaptation, reducing DER compared to models trained on narrowband data.

1.3.3 Voice Activity Detection (VAD) Postprocessing

While the TS-VAD+ model inherently performs speaker detection and voice activity detection (VAD), postprocessing with a specialized VAD model can refine segmentation accuracy. This work integrates pyannote's pre-trained VAD model to correct false alarms in speaker detection. Results indicate that VAD postprocessing reduces the overall DER by approximately 1.5% by effectively eliminating non-speech segments while maintaining high speaker detection accuracy.

1.3.4 Enhanced Clustering for Speaker Detection

Initial speaker embeddings used for TS-VAD+ training are typically extracted using x-vector clustering, which can lead to suboptimal speaker segmentation. This thesis re-

places x-vector clustering with VBx (Variational Bayes HMM clustering), which improves speaker segmentation before embedding extraction. The adoption of VBx clustering reduces the diarization error rate by approximately 1.88% and improves performance in multi-speaker conditions.

1.3.5 Memory Module for Speaker Embeddings (mm-TS-VAD+)

Standard TS-VAD+ relies on static speaker embeddings, which may fail to generalize well in unseen conditions. This thesis proposes Memory Module TS-VAD+ (mm-TS-VAD+), which incorporates a memory-aware attention mechanism to dynamically refine speaker embeddings. The memory module aggregates knowledge from a diverse corpus of speaker embeddings, enhancing speaker tracking capabilities. While mm-TS-VAD+ shows clear improvements when pretrained on simulated data, its benefits are limited by the size of the fine-tuning dataset, suggesting that larger training corpora could further enhance its effectiveness.

1.3.6 Data Augmentation for Robustness

A key challenge in diarization is the limited size of high-quality annotated datasets. To address this, this work applies a novel data augmentation strategy by mixing single-speaker and two-speaker recordings in the DIHARD III dataset while ensuring realistic overlapping speech conditions. This augmentation improves model robustness, leading to a 3.75% improvement in DER for mm-TS-VAD+ compared to training on the original dataset alone.

1.3.7 Performance Improvement on DIHARD III

Through systematic experimentation and iterative improvements, TS-VAD+ achieves a competitive benchmark in speaker diarization on the DIHARD III dataset (track 2). The final model, incorporating ECAPA-TDNN embeddings, VBx clustering, wideband pretraining, and VAD postprocessing reduces DER to 19.11%, surpassing baseline models such as VBx-HMM (21.97%) and EEND (20.05%).

Chapter 2

Literature Review

Various approaches for speaker diarization have been proposed over the years, from traditional clustering-based pipelines to deep learning-based approaches. Each method has its strengths and limitations that influence its suitability for different applications. In the below sections I have provided an overview on some of the most widely known speaker diarization systems. Firstly I will introduce Diarization Error Rate which is the error metric used to measure a systems performance.

2.1 Diarization Error Rate

The Diarization Error Rate (DER), first introduced by the National Institute of Standards and Technology (NIST) as part of Rich Transcription (RT) evaluations in the early 2000s, is a standard metric used to evaluate the performance of speaker diarization systems. It measures the percentage of time that a diarization system incorrectly assigns speech segments to speakers. DER accounts for three types of errors:

- False alarm: Time when the system incorrectly labels non-speech as speech.
- Missed speech: Time where speech was present but not labelled by the system.
- Speaker confusion: Time where a speaker is mislabeled

The metric is computed as the sum of these errors, typically expressed as a percentage of the total speech duration in an audio recording.

$$\% \text{ DER} = \frac{(\text{False alarm} + \text{Missed speech} + \text{Speaker confusion})}{\text{Total time of speech}} \times 100$$

2.2 Clustering-Based Diarization System

Clustering-based diarization approaches are the traditional method for speaker diarization, where a pipeline approach is used. An overview of the pipeline can be seen in Figure 2.1.

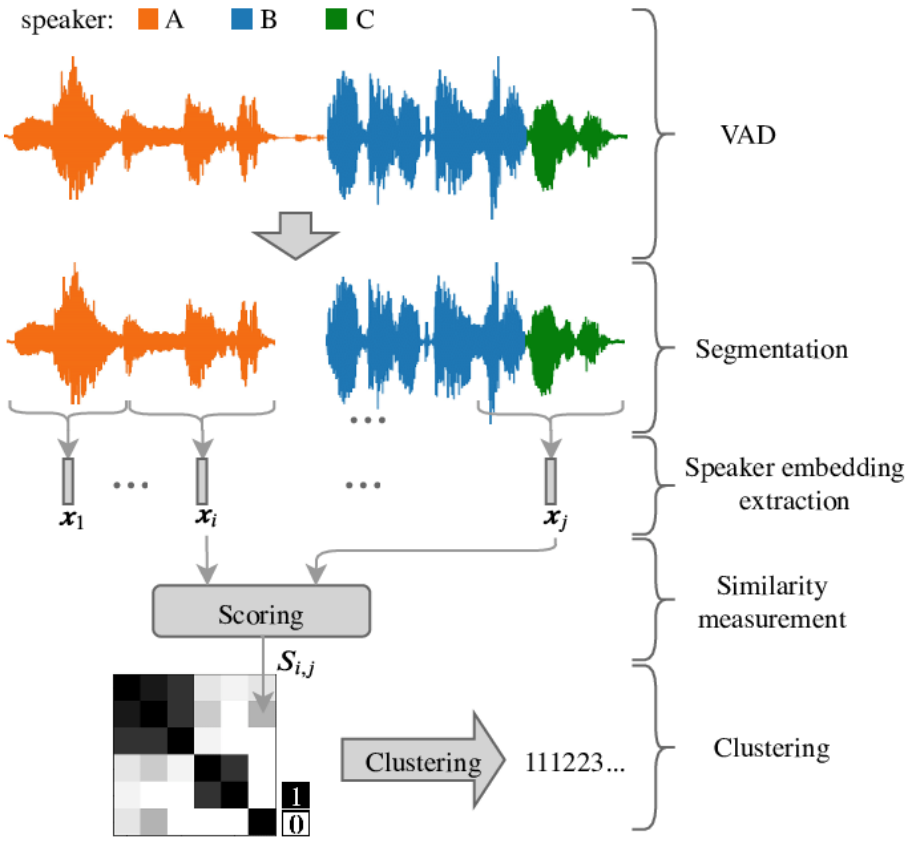

Figure 2.1: Clustering-based diarization pipeline [24].

Architecture:

1. Voice Activity Detection (VAD): The input audio is processed through a VAD model which separates the speech segments from the non-speech regions (silence, background noise, acoustic disturbances). Some VAD models are built on energy-based or statistical modelling techniques, while newer VAD models rely on RNNS, CNNs or other deep learning architectures. An effective VAD model ensures that only speech frames are sent forward so that the missed speech error is as low as possible, improving the accuracy of the overall model.

2. Preprocessing: Further preprocessing can be applied to the audio, such as overlap speech detection (OSD) or Speaker Change Detection. OSD models label the re-

gions with overlapped speech separately from single-speaker speech. In contrast, speaker change detection detects and labels the speech regions where the speaker identity changes. This extra information into further steps enhances the performance of clustering algorithms.

3. Feature Extraction: The audio is then processed to extract acoustic features in the form of Mel-Frequency Cepstral Coefficients [37]. They are derived from the short-term power spectrum of speech, designed to mimic the human auditory system by emphasizing lower frequencies where most speech information is concentrated. These MFCCs act as a lower-dimensional representation of speaker characteristics and can be used as input for further steps. More advanced systems use newer and more robust representations such as i-vectors (generated using statistical models on MFCCs) or x-vectors [**xvectors**] (generated using deep learning models on MFCCs). These outperform MFCCs in cases with more noise and channel variations at the cost of computational complexity.

4. Clustering: The extracted features are then clustered based on similarity. Standard clustering algorithms used are Agglomerative Hierarchical Clustering, which merges clusters based on similarity; Spectral Clustering (SC) [27, 17], which constructs an affinity matrix and applies clustering in an embedding space; and K-Means Clustering works by assigning data points to the closest cluster centroid, iteratively updating the centroids to minimize the overall distance within clusters.

5. Post-processing: The direct output from clustering has lots of incorrect segment boundaries; hence, to enhance robustness to variations in speaker identity, short speech segments and background noise, probabilistic models such as Hidden Markov Models and Gaussian Mixture Models are applied. E.g. the VB-HMM resegmentation model [23], accounts for uncertainty in speaker assignments by iteratively updating the posterior probability of each segment belonging to a speaker through optimization of a Bayesian Objective Function. Speech segments are reassigned using scoring functions that compute the likelihood that two embeddings belong to the same or different speakers.

2.3 End-to-End Neural Diarization (EEND)

End-to-end neural Diarization (EEND) was first introduced by Fujita et al. [13] in 2019 as a deep learning-based alternative to clustering-based diarization methods. Unlike the step-by-step pipeline approach of first extracting features and then processing, clustering and resegmenting the output for final results, EEND does all the steps in one go using a transformer-encoder-based architecture by utilizing the self-attention mechanism of transformers for capturing long-range dependencies in the data A.1. One limitation of EEND was that it only worked with a fixed number of speakers. An overview of the architecture can be seen in the figure below:

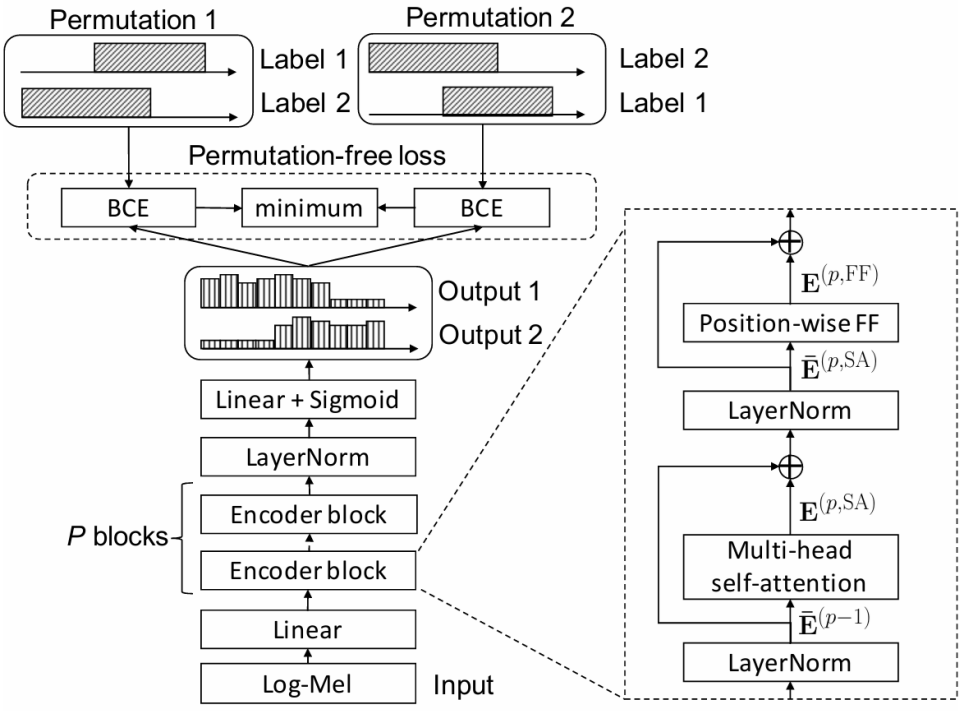

Figure 2.2: Example of a 2-speaker EEND model architecture [13].

Architecture:

1. Feature Extraction: The raw audio input is converted to spectrogram-based features such as MFCC's or log Mel filterbank features [31].

2. Transformer Encoder: The extracted features are used as input for a multi-layer transformer encoder. This transformer encoder consists of a multi-head self-attention block as well as feed-forward layers. Layer normalization and dropout functions are used in between layers to stabilize training and prevent overfitting. Detailed overview is given at A.2.

3. Diarization Output Layer: The transformer encoder output is then passed through a fully connected neural network with a sigmoid activation function to predict the speaker presence probabilities for each frame. The model uses permutation invariant training to reorder output speaker labels to minimize the diarization loss (A.3). This ensures that the model's loss function is independent of speaker order since the model does not assume a fixed speaker order. .

Improvements were made to the original EEND architecture, such as the EEND-EDA (Encoder-Decoder Attractor) [20] that allowed for a variable number of speakers. It introduced a speaker attractor mechanism, which enabled the model to determine the number of active speakers in an audio segment rather than rely on a predefined speaker count. EEND-VC (EEND with vector clustering) [9] was another proposed system which combined the transformer encoder mechanism in EEND with clustering. The transformer model was used to extract speaker embeddings clustered to group similar embeddings into speaker identities. This hybrid approach combined the robustness of deep learning-based approaches with maintaining the model's adaptability to varying speaker counts.

2.4 Target-Speaker Voice Activity Detection (TS-VAD)

The original TS-VAD model, proposed by Medennikov et al. [25], is a deep learning-based approach focusing on detecting target speakers' speech activity by leveraging pre-enrolled speaker embeddings. It was developed specifically for scenarios involving overlapping speech with more speakers, as other approaches struggled in these scenarios.

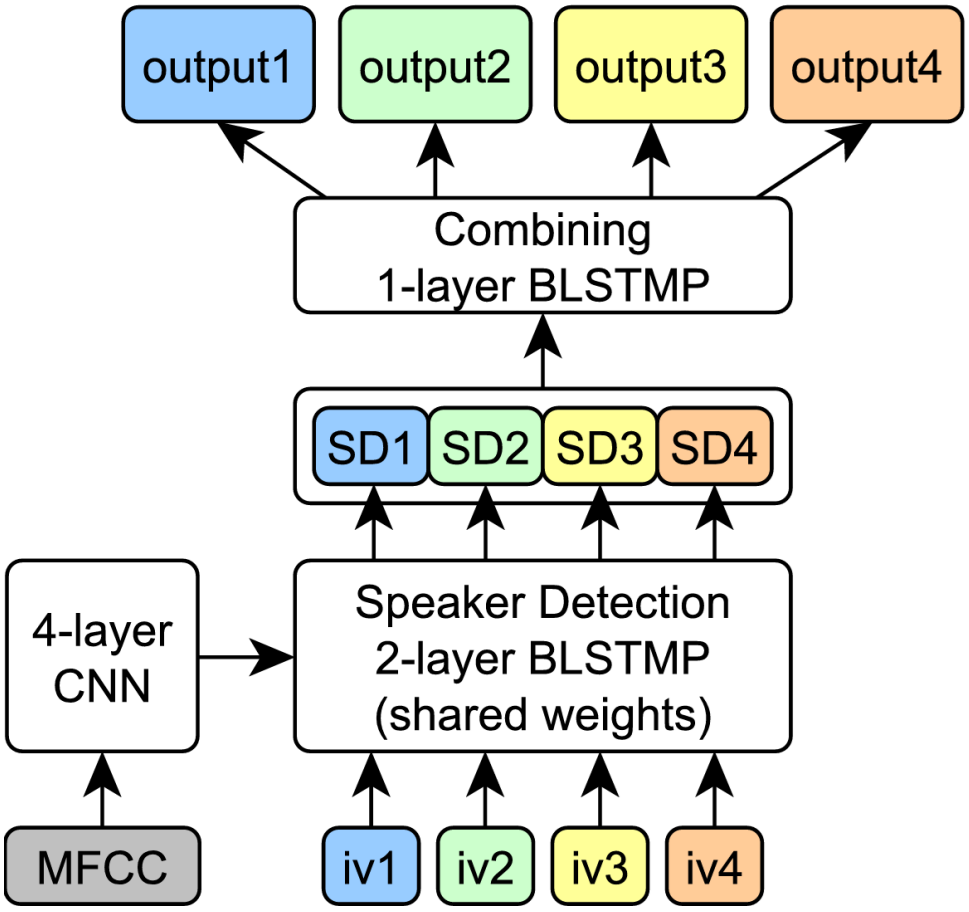

Figure 2.3: TS-VAD Model Architecture [25].

Architecture:

1. Input representation

    - Speech Features: The first kind of input to the model consists of frame-wise spectrogram-based features (MFCCs). These features are generated from the raw audio input over short overlapping frames (25ms long frames with 10ms overlap is a common choice) to preserve temporal resolution. A simple CNN-based feature extraction method extracts speech features from the MFCCs.
    - Speaker Embeddings: Speaker embeddings are fixed-dimensional compact numerical representations that capture the characteristics of each speaker. The TS-VAD model uses i-vectors as input speaker profiles. I-vectors are speaker embeddings generated using a Gaussian Mixture - Universal Background Model (GMM-UBM). These are primarily used in speaker recognition and speaker diarization tasks.

2. Neural network processing

    - Frame-Level Concatenation: The fixed-length speaker embeddings are replicated and concatenated with the speech features at every frame. This ensures that both the spectral information (helping determine whether speech is present) and the speaker identity information (to determine which speaker is active) are being passed to the neural network.
    - Bidirectional LSTM: The concatenated input is passed through a Bidirectional Long Short-Term Memory (BLSTM) network. This layer captures the long-range dependencies in the audio. It assigns speaker labels by taking information regarding speaker identity from the embeddings and matching that with the audio input to predict the likelihood of each speaker being active at a given time. The output of this layer is in the form of a frame-level probability for each speaker in the audio.
    - Refinement: A second BLSTM layer is applied to aggregate the probabilities from the previous stage while enforcing temporal consistency in speaker assignment. The final output of this layer consists of a single frame-level probability matrix. These can be post-processed to generate a Rich Transcription Time Marked (RTTM) file (A.4).

The above models, although innovative, had glaring drawbacks. For clustering-based systems, because of the pipeline approach, errors in the VAD model, OSD model or speaker change detection model are propagated throughout the process. Clustering works well only in single-speaker scenarios, as it cannot handle overlapping speech.
The EEND models tried to overcome this functionality gap but lost efficiency as enormous amounts of data are required to train the model effectively. The model's performance is highly dependent on the domain of the training data and the eval data. If there is a domain mismatch, the model performs much worse, highlighting its lack of robustness

in different scenarios. Since the model does not store explicit speaker embeddings, it cannot track speakers across multiple sessions or meetings, making it less suitable for applications requiring long-term speaker identification.
Even the TS-VAD model had significant drawbacks, as it can only work for a fixed number of speakers because of its architecture. Looking at these weaknesses, the TS-VAD+ model was built.

2.5 Overview of Diarization systems throughout the years

Model	**Year**	**DER%**	**Description**
VBx-HMM + OSD	2020	21.97%	A clustering-based approach that uses VBx clustering combined with Overlapped Speech Detection postprocessing to refine speaker segmentation
EEND-EDA	2022	20.69%	End-to-end neural diarization with encoder-decoder architecture and speaker attractors
DiaPer	2024	20.3%	End-to-end neural diarization using Perceiver-based attractors, improving performance and inference speed.
NTU's TS-VAD+	2024	19.10 %	A transformer-based TS-VAD model integrating improved speaker embedding and speech encoding, along with VAD postprocessing
Mamba-EEND-VC	2024	16.7%	Utilizes the Mamba architecture, which combines RNN-like behavior with attention mechanisms, to enhance speaker diarization performance
ANSD-MA-MSE	2024	16.76%	Neural speaker diarization with memory-aware multi-speaker embedding, improving performance in challenging acoustic conditions.

Table 2.1: Roadmap of Speaker Diarization Systems and their performance on DIHARD III eval set (track 2)

Chapter 3

NTU's TS-VAD+ Model

This chapter introduces TS-VAD+, an improved transformer-based speaker diarization model developed to address the limitations of existing approaches. The chapter details key enhancements including the integration of ECAPA-TDNN for speaker embeddings, a WavLM-based speech encoder for feature extraction, and a transformer-based architecture for speaker activity detection.

A major challenge in speaker diarization is the robustness of speaker embeddings, particularly in unseen environments. To address this, we introduce the Memory Module TS-VAD+ (mm-TS-VAD+), which incorporates a memory-aware attention mechanism to refine speaker embeddings using external speaker representations. We describe the construction of the memory module, its integration into TS-VAD+, and the attention mechanism that enables adaptive speaker representation learning.

Additionally, this chapter outlines the pretraining strategy for TS-VAD+, detailing the use of wideband simulated data to improve domain adaptation. The training pipeline, including data preprocessing, augmentation strategies, and inference mechanisms, is also covered. These improvements collectively enhance TS-VAD+'s ability to handle complex multi-speaker scenarios with high-overlap speech.

3.1 Idea

The TS-VAD+ model was created by improving upon each key component of the original TS-VAD model.

- Speaker embedding model: Instead of an outdated i-vector embeddings, the idea was to use pre-trained speaker embeddings models whose architecture is innovatively designed to capture the speaker characteristics even with short audio length or noisy speech scenarios. Models such as ECAPA-TDNN, CAM++, ERes2Net are some state of the art approaches for generating speaker embeddings [10, 36, 7]. The TS-VAD+ model uses the ECAPA-TDNN model for speaker embeddings.

- Feature extraction: Instead of a comparatively naive approach of CNN speech encoder, the TS-VAD+ model uses WavLM speech encoder model [6]. WavLM is a self-supervised speech encoder developed by Microsoft that builds upon Wav2Vec 2.0 and HuBERT [21, 3]. It is particularly designed for applications like automatic speech recognition (ASR), speaker diarization, speech enhancement, and emotion recognition as it is pre-trained with noisy speech and multi-speaker scenario data.
- Speaker detection layers: The TS-VAD+ uses the Transformer Encoder architecture for labeling the speaker identities within speech segments and another Transformer layer for combining the data for all speakers into one set of probabilities for each frame of speech [35]. Then basic postprocessing is used to convert the output vectors into rttm files.

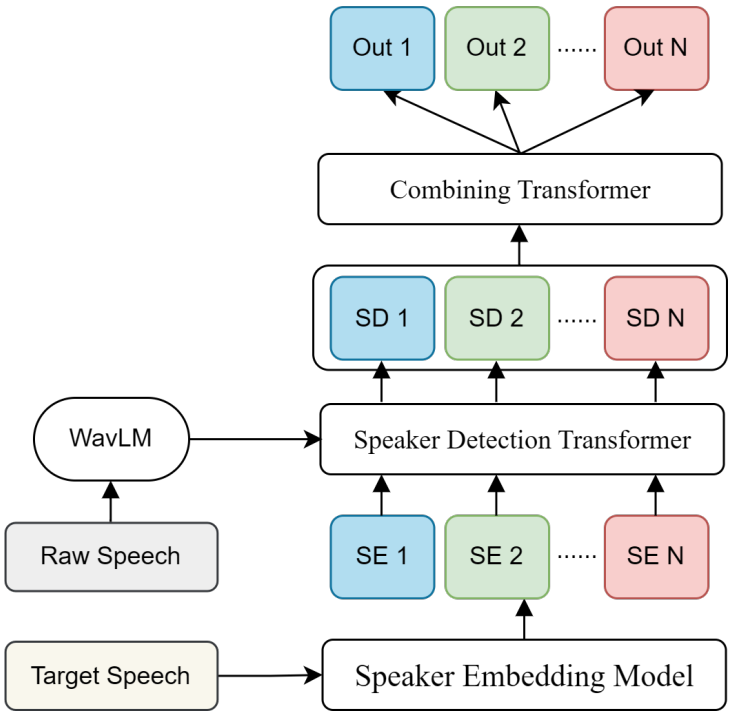

Figure 3.1: TS-VAD+ Model Architecture [25].

3.2 Architecture:

The TS-VAD+ model is designed to handle up to N speakers in a given audio segment. The feature extraction module utilizes a WavLM-based front end, which processes raw audio waveforms and extracts frame-level embeddings. These embeddings are represented as a tensor $E \in \mathbb{R}^{T \times E}$, where T denotes the sequence length and E is the embedding

dimension. The extracted embeddings capture rich speaker and phonetic information, crucial for diarization in overlapping speech conditions.

The speaker encoder module leverages the ECAPA-TDNN model to generate speaker profiles. Given an input audio segment with N target speakers, the speaker encoder extracts N corresponding speaker embeddings P_n, each of dimension P. These speaker profiles are then duplicated T times and concatenated with the frame-level embeddings E, producing N speaker-conditioned embeddings $Q_n \in \mathbb{R}^{T\times(E+P)}$. These embeddings serve as inputs to the independent-speaker-detection module, enabling the model to process speaker information in a structured manner.

The independent-speaker-detection module consists of six transformer layers, each operating separately on the N mixed embeddings. These layers capture temporal dependencies and refine speaker-specific information, outputting N feature representations $F_n \in \mathbb{R}^{T\times F}$, where F denotes the new feature dimension. The outputs from all N speaker branches are then concatenated into a single tensor of shape $\mathbb{R}^{T\times(N\cdot F)}$, integrating speaker-specific and temporal features into a unified representation.

A second transformer-based module is then applied to this combined feature representation to detect the speech activity of each target speaker. The model produces a binary output tensor of shape $T \times N$, indicating the presence or absence of each speaker at every time frame. The training objective is to minimize the sum of binary cross-entropy losses across all N speaker activity outputs, ensuring robust diarization even in high-overlap conditions.

One of the key challenges in training TS-VAD+ is selecting the target speakers for embedding extraction. During training, ground-truth labels are used to extract embeddings for the active speakers in each segment. However, when the number of active speakers in a segment is fewer than N, two strategies are employed to maintain consistency in input dimensionality:

- Augmentation Strategy: Additional speaker profiles are randomly sampled from other utterances to artificially increase the number of inputs to N.
- Zero-padding Strategy: If augmentation is not used, the missing speaker slots are padded with zero tensors, ensuring the expected input dimensionality is preserved.

During inference, a clustering system (VBx or x-vector clustering) is used to determine the target speakers from the given audio. Unlike training, where ground-truth labels are available, inference relies on clustering to estimate speaker identities.

This enhanced TS-VAD+ framework, with pretrained speaker embedding models, multi-stage transformers, and a robust inference mechanism, significantly improves speaker diarization performance, particularly in scenarios with high speaker overlap, as demonstrated in the DIHARD III (Track 2) experiments.

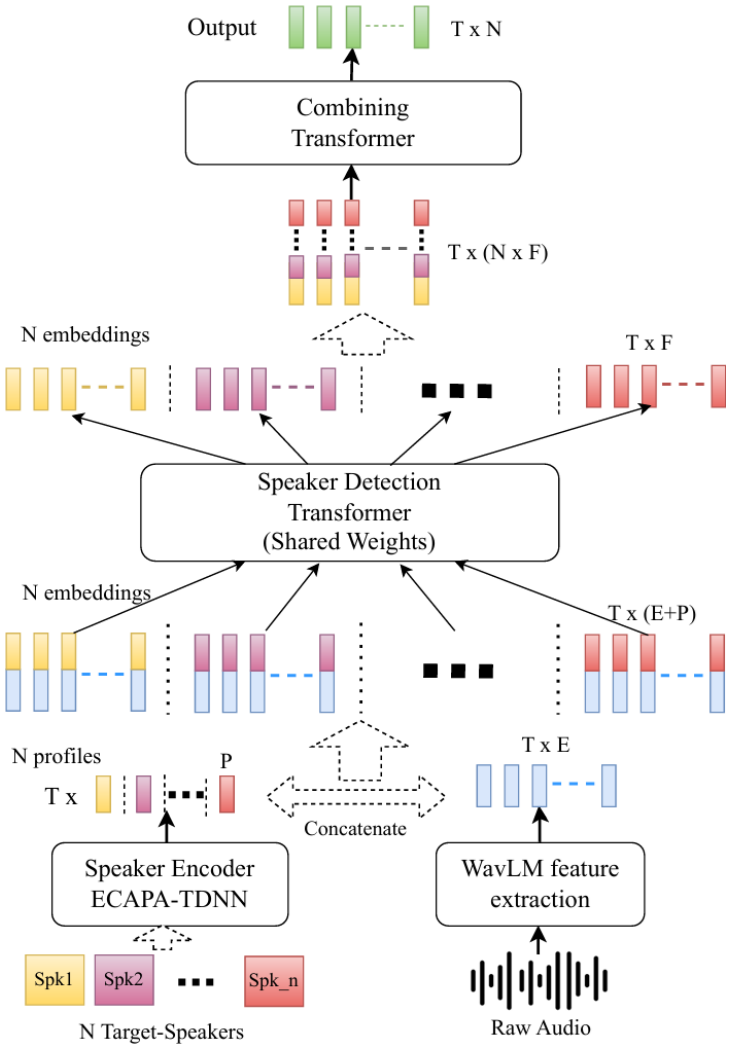

Figure 3.2: TS-VAD+ Model dataflow [1].

3.3 Memory Module TS-VAD+

The original TS-VAD as well as the TS-VAD+ model relies on speaker embeddings to be able to characterize the atributes of different speakers voices, hence the quality of these embeddings directly affects diarization accuracy. Moreover, speaker embeddings derived from a limited dataset or short audio segments may not adequately capture the full variability of speaker traits. Recent research has shown that by providing additional speaker context to these embeddings through special attention mechanisms, the quality of these embeddings can be enhanced . In this section, we propose a Memory Module TS-VAD (mm-TS-VAD) architecture, inspired by the memory-aware attention mechanism introduced by [19]. The idea is to create a memory module from external speaker embeddings,

which will act a learned memory component. By introducing a memory module we seek to (1) provide a global context of speaker attributes gathered from a large corpus, and (2) allow embeddings to dynamically refine themselves by selectively integrating information from the memory module using a specially designed attention mechanism.

3.3.1 Memory Module Construction

To construct the memory module (MM), first collect a large corpus of speaker embeddings. This allows the MM to represent a wide range of speakers and acoustic conditions. We then apply k-means clustering to these embeddings, resulting a set of cluster centroids. Each cluster centroid can be viewed as a "memory slot" that captures a dominant configuration of speaker characteristics. For our task we used to already provided cluster centroids in the NSD-MA-MSE github repo [19]. They were made using the VoxCeleb 1 and 2 datasets by first creating i-vector embeddings from the audios in these datasets and then performing k-means clustering with different number of clusters (K = 64, 128 and 256). Same process was repeated by using x-vector embeddings instead. Hence a total of 6 memory modules were created.

3.3.2 Memory-Aware Attention Mechanism

The memory-aware attention mechanism is designed to refine speaker embeddings by integrating additional context from a memory module to improve the overall model performance.

Initially, frame-wise speech features are extracted using a speech encoder model (WavLM in our case). Speaker label information from the input RTTM files is then utilized to retain only the speech features corresponding to active speaker frames, while all other frames are zeroed out. This selection process is facilitated by a speaker mask matrix, ensuring that only relevant speaker-specific information is preserved as input to the memory-aware attention mechanism.

Memory-aware attention: Firstly a linear transformation is applied on the speech features (x) and the memory module tensors (m) using linear mappings W and U respectively. Then using a similarity metric, a similarity score is calculated for each cluster centroid (c_k). A higher score for a given centroid indicates that the speaker's characteristics are more aligned with that cluster.

This score is normalised using a sigmoid function to obtain the attention weights (a_k). These weights reflect the degree of similarity between the speaker's features and each memory centroid. Using these attention weights, we compute a weighted sum of the memory centroids. This weighted sum serves as an enhanced speaker embedding, where centroids that are more similar (i.e., receive higher weights) have a higher contribution to the refined embedding, thereby enriching the original embedding with broader, context-

driven speaker information.

$x \in \mathbb{R}^{F \times T}$ is the input speaker feature tensor, where F is the feature dimension, T is the number of time frames.

$m \in \mathbb{R}^{K \times E}$ is the memory modules where K is the number of cluster centroids, E is the dimension of each centroid which is also the dimension of the embeddings from which the memory module was created.

$W : \mathbb{R}^F \to \mathbb{R}^H$ and $U : \mathbb{R}^E \to \mathbb{R}^H$ are learnable linear mappings to an H-dimensional space, where H is the number of attention heads. Let $v : \mathbb{R}^H \to \mathbb{R}$ be a linear projection that outputs a scalar score.

Then the similarity score $c_k \in \mathbb{R}$ and subsequently the attention weights are computed for each input tensor, defined as:

$$c_k = v(\tanh(W(x) + U(m))$$

$$a_k = \sigma(c)$$

Finally, a weighted sum over the memory centroids is computed to yield the enhanced embedding:

$$e = \sum_{k=1}^{K} a_k \; m_k$$

where $e \in \mathbb{R}^E$ and m_k represents the k^{th} cluster centroid.

3.3.3 Integration into the TS-VAD+ model

The enhanced embedding for a given input speaker vector is concatenated with the originally generated speaker embedding and subsequently downsampled to restore its original dimensionality, yielding a fused representation. This merged embedding retains the intrinsic characteristics of the original speaker representation while incorporating additional contextual information from the memory module. The resulting speaker profile is then provided as input to the TS-VAD+ model, with the remainder of the processing pipeline remaining unchanged.

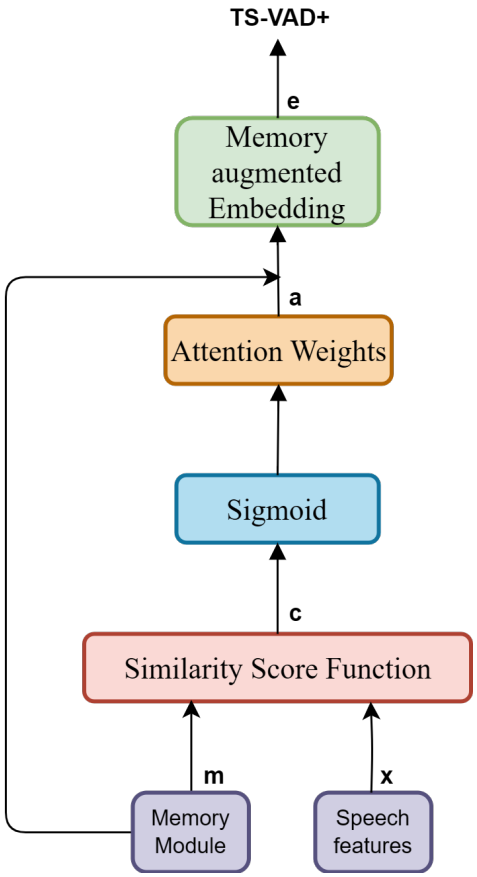

Figure 3.3: Memory Aware Attention Architecture

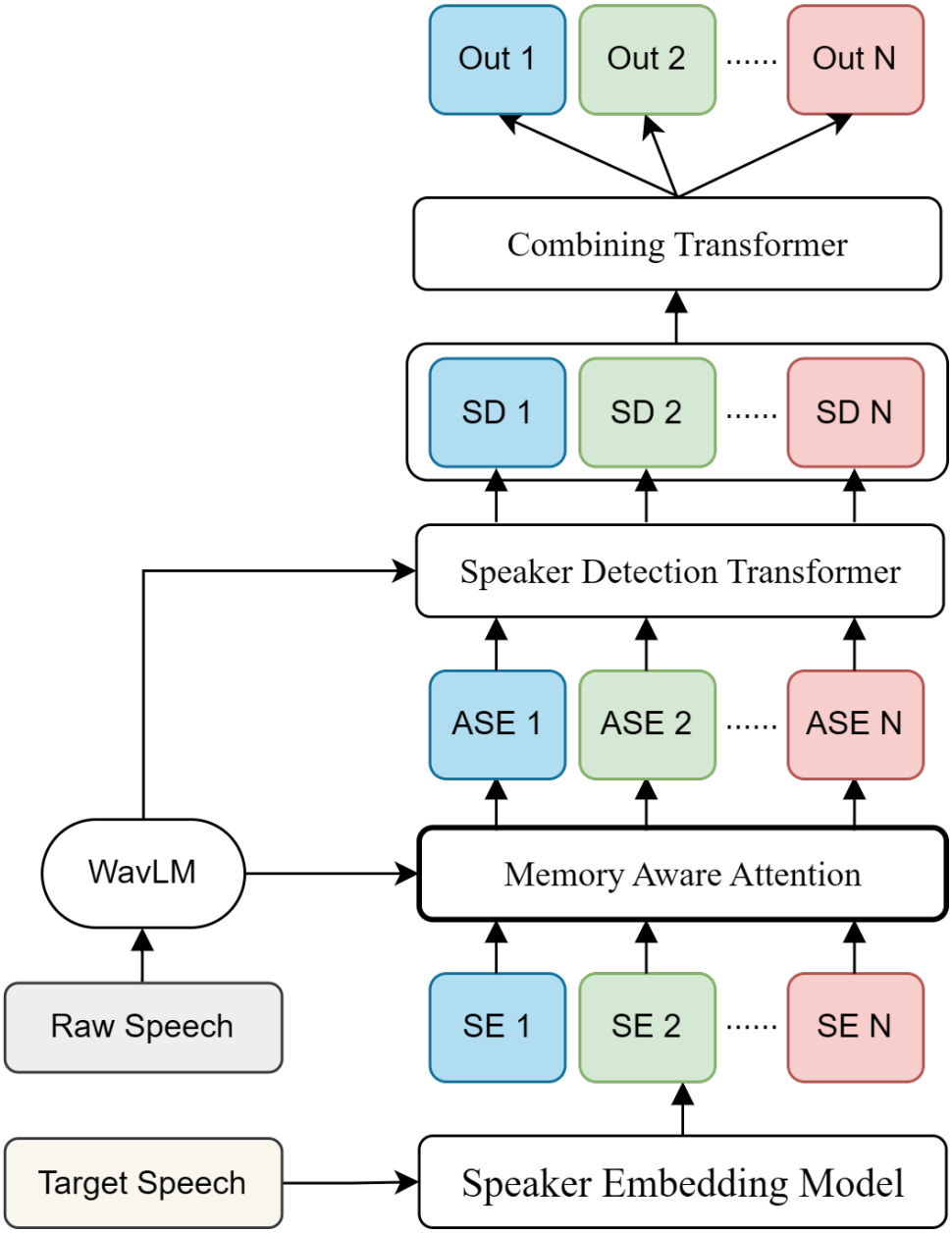

Figure 3.4: mm-TS-VAD+ Model Architecture [25].

Chapter 4

Results and Discussion

This chapter dives into the evaluation of the proposed Target Speaker Voice Activity Detection (TS-VAD) framework. It details the experimental setup, including datasets, training procedures, and evaluation metrics. The chapter then presents the results, demonstrating that TS-VAD achieves competitive performance in speaker diarization tasks.

4.1 Datasets

4.1.1 DIHARDIII dataset

The DIHARDIII [32] Dataset is a meticulously curated collection of audio sources, designed to advance speaker diarization technologies. It includes a diverse range of audio environments such as monologues, interviews, YouTube videos, and conversational telephone speech from the Fisher English Phase 2[8] collection. The dataset also consists of 5-10 minute segments sampled from 11 conversational domains, covering a wide range of recording equipment, environments, noise levels, speaker demographics, and interaction types.

The DIHARDIII challenge focuses on 'hard' diarization, aiming to tackle scenarios where current technologies fall short. It evaluates performance using two primary metrics, the Diarization Error Rate (DER) and Jaccard Error Rate (JER), with no forgiveness collar applied and overlapping speech evaluated. The Third DIHARD Challenge includes two tracks: diarization from reference Speech Activity Detection (SAD) and diarization from scratch. It does not provide a fixed training set, allowing participants the freedom to train their systems on any proprietary and/or public data. This open approach fosters a collaborative and inclusive environment for innovation in the field of speaker diarization.

4.1.2 Narrowband Simulated Data

For my initial experiments, the model was trained using simulated speech mixtures that are publicly available as part of the EEND repository developed by Yusuke et al. (2019)

[13].These mixtures were created from the Switchboard[14], the Switchboard Cellular (Phases 1 and 2)[15, 16], and the NIST Speaker Recognition Evaluation corpora[29] following the protocol described in their paper. Briefly, the protocol involves selecting N speakers, randomly sampling speech segments, convolving them with room impulse responses, and mixing them with noise at varying signal-to-noise ratios. The simulation protocol also offers flexibility in controlling parameters such as the overlap ratio among speakers by adjusting the mean value parameter (β) of the exponential distribution governing the duration of silence between speech segments, ensuring a consistent and customizable simulation environment across different speaker configurations. The drawback of this simulated data was that it had a frequency of 8kHz. It was upsampled to 16kHz using the 'sox' tool. This step was necessary as the WavLM model is trained on 16kHz audio, and both the DIHARD development and evaluation sets are also in 16kHz format.

4.1.3 Wideband Simulated Data

During the course of my thesis, a new set of wideband simulated speech mixtures was developed by my mentor, Tran The Anh, to better align with the target datasets. These mixtures were created using speech samples from the LibriSpeech corpus [28], which consists of read English speech recordings. The simulation process, referred to as Simulated Conversations (SC), was designed to closely mimic natural speaker interactions by leveraging reference RTTM files to determine speaker turns, silence durations, and overlapping speech patterns, thereby creating realistic conversational dynamics. To further enhance realism, room impulse responses (RIRs) were applied to simulate reverberation effects, and background noise from the MUSAN corpus [34, 22] was added at varying signal-to-noise ratios. Unlike the previously used simulated data, this dataset was generated at a sampling rate of 16kHz, eliminating the need for upsampling and ensuring consistency with the WavLM model and the DIHARD evaluation framework.

4.2 Training Procedure

The experimental setup commenced by employing a clustering model (such as x-vector clustering or VBx) on the evaluation dataset to acquire speaker labels for each utterance and process them into RTTM files. Using speaker turn information from these RTTM files, the input audios were segmented by removing overlapping speech segments. The remaining audio segments for each speaker were concatenated and saved as individual files, referred to as target audios. For the development set, ground truth RTTM files were used to create the target audios. However, for the evaluation set, where ground truth annotations were unavailable, clustering output RTTM files were utilized to generate target audios.

Subsequently, the audio was split into shorter segments for embedding extraction. Speaker embeddings were computed using pretrained speaker embedding models. Target audios

and embeddings were created for the n longest speakers in each utterance, where n represents the maximum speaker capacity of the TS-VAD+ model. In cases where the number of speakers was fewer than the model's maximum speaker count, zero tensors were used for padding. To ensure consistency in the dataset, data augmentation techniques were applied. The training pipeline incorporated Microsoft's WavLM Base Plus model for speech encoding. The TS-VAD+ model was initially pretrained on simulated data for 40 epochs, with a warm-up period of 10 epochs, before being fine-tuned on the DIHARDIII development set for an additional 25 epochs. A batch size of 40 was used during training. During pretraining with the narrowband simulated data, the DIHARDIII dataset was first downsampled to 8 kHz and subsequently upsampled to 16 kHz. This preprocessing step was implemented to align the spectral characteristics of the DIHARDIII data with those of the simulated dataset, which was originally 8 kHz and later upsampled to 16 kHz. The rationale behind this approach was to ensure consistency in the frequency signature between the pretraining and fine-tuning datasets, for better domain adaptation. Data augmentation techniques including noise augmentation from the MUSAN dataset, which consists of music, speech, and noise, along with Room Impulse Response (RIR) augmentation were implemented. These augmentations introduced various types of background noise and reverberations to simulate real-world scenarios, enhancing the model's robustness and generalization capability across diverse acoustic environments.

The loss function employed during training was BCEWithLogitsLoss, aimed at minimizing the binary cross-entropy between the model's predictions and the ground truth labels. The TS-VAD+ model's performance was evaluated using the Diarization Error Rate (DER), which was calculated using the md-eval Perl script with no collar. This structured training approach ensured that the TS-VAD+ model effectively learned speaker diarization patterns while adapting to diverse and challenging real-world conditions.

4.3 Experimental results

4.3.1 Evaluation of Speaker Embedding Models for TS-VAD+

A crucial aspect of optimizing the TS-VAD+ model for speaker diarization is selecting an appropriate speaker embedding model. This evaluation was performed using the DIHARDIII dataset (track II) without pre-training on simulated data. Furthermore, to maintain consistency and allow for a direct comparison, all other components of the TS-VAD+ model were kept identical, and the maximum number of speakers (max-speaker parameter) was set to 8. Various pre-trained speaker embedding architectures have been developed, each offering distinct advantages in capturing speaker characteristics. We conducted a comparative analysis of three widely used architectures: CAM++, ECAPA-TDNN, and ERes2Net. Through a systematic evaluation of these three architectures, we aim to determine the most effective speaker embedding model for TS-VAD+, which will serve as the foundation for subsequent experiments.

Model	% DER (8 spk)
ERes2Net	25.34
CAM++	23.42
ECAPA-TDNN	**22.96**

Table 4.1: DER score comparison with different speaker embedding models

Based on this comparison, we concluded that the ECAPA-TDNN speaker embedding model is the most suitable for our architecture.

4.3.2 Comparison of TS-VAD+ models

With the optimal speaker embedding architecture identified, the next step was to evaluate the model's performance on the DIHARDIII (Track 2) dataset while varying the maximum number of speakers. Using ECAPA-TDNN as the speaker embedding model, three versions of TS-VAD+ were trained on narrowband simulated data and subsequently fine-tuned on the DIHARDIII dev set, differing only in the max_speaker parameter, which was set to 4, 6, and 8, respectively. Performance was assessed both at the domain level and across the entire dataset, with x-vector clustering results included for reference.

In 4.2, bold values indicate the best score for each domain, highlighting the model configuration that achieved optimal diarization performance in different scenarios. The results indicate that TS-VAD+ consistently outperformed x-vector clustering in domains with high-overlap speech (e.g., conversational telephone speech, meetings, restaurants, and web videos). Notably, the configuration with a max_speaker value of 6 yielded the best overall performance.

4.3.3 Wideband simulated data

New wideband simulated data was generated using librispeech and dynamic speaker mixing techniques. Unlike the previously used simulated data, this dataset was generated at a sampling rate of 16kHz, eliminating the need for upsampling and ensuring consistency with the WavLM model and the DIHARD evaluation framework. So a comparison of the wideband vs narrowband simulated data was done. One model was pretrained on narrowband simulated data then finetuned on DIHARDIII dev set. The other model was pretrained on wideband simulated data then finetuned on DIHARDIII dev set. Both models used ECAPA-TDNN for speaker embedding extraction and were evaluated on DIHARDIII eval set, with max_speaker set to 8.

The results indicate that models pretrained on wideband simulated data (16kHz) consis-

ID	Domain	No. of spkr	Overlap Ratio	x-vector clust	TS-VAD+ (4 spk)	TS-VAD+ (6 spk)	TS-VAD+ (8 spk)
1	audiobook	1	0.00%	**3.04**	8.90	9.09	9.23
2	broadcast int.	3-4	1.61%	**10.05**	14.57	14.98	14.19
3	clinical	2-3	3.13%	25.11	24.23	23.50	**23.30**
4	court	3,6-9	1.79%	**15.86**	24.77	21.01	21.00
5	cts	2	10.92%	19.8	13.95	**13.82**	14.48
6	maptask	2	1.83%	**11.15**	16.99	15.30	15.63
7	meeting	3-6	21.30%	43.42	43.57	**41.74**	43.59
8	restaurant	4-8	26.61%	59.1	51.91	51.37	**51.33**
9	socio field	2-3	4.53%	20.98	16.46	**16.27**	17.71
10	socio lab	2	3.58%	**12.63**	13.99	12.83	13.00
11	webvideo	1-7	17.31%	53.85	50.20	50.35	**49.98**
	Total	1-9	8.75%	24.99	23.19	**22.57**	22.96

Table 4.2: Domain wise DER score comparison of different TS-VAD+ models

tently achieve lower DER scores across most domains compared to those pretrained on narrowband simulated data (8kHz upsampled to 16kHz). This supports our hypothesis that using 16kHz simulated data for pretraining is beneficial. Since WavLM is pretrained on 16kHz data and both the DIHARDIII dev and eval sets contain 16kHz recordings, maintaining the original sampling rate during simulation leads to better domain adaptation, as it ensures a closer alignment with the real-world data characteristics.

4.3.4 Improvements due to better Clustering

After establishing that the wideband simulated data was suitable for our task, further improvements were pursued by incorporating a more refined clustering approach. A new VBx clustering pipeline was developed by a lab member, yielding improved clustering results compared to the initial x-vector clustering approach. Leveraging these enhanced clustering outputs, we reprocessed the evaluation target audios and observed a further reduction in Diarization Error Rate (DER) [23]. The training procedure remained consistent, with models initially pretrained on wideband simulated data before being fine-tuned on the DIHARDIII development set, while using ECAPA-TDNN for speaker embedding extraction. The max_speaker parameter was set to 8.

ID	Domain	No. of spkr	Overlap Ratio	TS-VAD+(8 spk) (narrowband)	TS-VAD+(8 spk) (wideband)
1	audiobook	1	0.00%	9.23	**8.75**
2	broadcast int.	3-4	1.61%	14.19	**14.17**
3	clinical	2-3	3.13%	23.30	**23.07**
4	court	3,6-9	1.79%	21.00	24.57
5	cts	2	10.92%	14.48	**14.07**
6	maptask	2	1.83%	15.63	**15.41**
7	meeting	3-6	21.30%	43.59	**37.85**
8	restaurant	4-8	26.61%	51.33	**51.02**
9	socio field	2-3	4.53%	17.71	**17.45**
10	socio lab	2	3.58%	13.00	**12.45**
11	webvideo	1-7	17.31%	**49.98**	50.13
	Total	1-9	8.75%	22.96	22.53

Table 4.3: Domain wise DER score comparison of TS-VAD+ models pretrained on different simulated data

In 4.4, bold values indicate the best score for each domain, highlighting the model configuration that achieved optimal diarization performance in different scenarios. The VBx clustering approach outperformed x-vector clustering, achieving approximately 1.5% lower DER. This improvement was reflected in the TS-VAD+ model, where the DER decreased from 22.53% to 20.65%, marking a significant reduction of 1.88%.

ID	Domain	No. of spkr	Overlap Ratio	x-vector clust	VBx clust	TS-VAD+(8 spk) (x-vector results)	TS-VAD+(8 spk) (VBx results)
1	audiobook	1	0.00%	**3.04**	4.10	8.75	8.97
2	broadcast int.	3-4	1.61%	10.05	**9.66**	14.17	14.57
3	clinical	2-3	3.13%	25.11	23.39	23.07	**21.57**
4	court	3,6-9	1.79%	15.86	**11.24**	24.57	11.40
5	cts	2	10.92%	19.8	18.34	14.07	**12.30**
6	maptask	2	1.83%	**11.15**	13.40	15.41	16.85
7	meeting	3-6	21.30%	43.42	40.64	37.85	**31.42**
8	restaurant	4-8	26.61%	59.1	54.85	51.02	50.88
9	socio field	2-3	4.53%	20.98	19.77	**17.45**	17.49
10	socio lab	2	3.58%	**12.63**	15.58	12.45	12.73
11	webvideo	1-7	17.31%	53.85	49.00	50.13	**48.12**
	Total	1-9	8.75%	24.99	23.41	22.53	**20.51**

Table 4.4: Domain wise DER score comparison using different clustering results

4.3.5 Postprocessing using Voice Activity Detection (VAD)

In the TS-VAD+ framework, Voice Activity Detection (VAD) and speaker labeling are performed simultaneously using a transformer architecture. While VAD models are specifically trained to identify speech regions, they often excel in this task due to their focused objective. To enhance the performance of TS-VAD+, we incorporate a VAD model as a post-processing step. After obtaining the initial results from TS-VAD+, the VAD model refines the segmentation by accurately identifying non-speech regions. This refinement reduces the false alarm rate by correcting mislabeled non-speech segments. However, this approach may lead to a slight increase in the missed speaker rate, as some speech regions might be incorrectly labeled as non-speech due to inherent limitations of the VAD model. Overall, it was assumed that the improvement due to the decrease in false alarm rate would be more than the detriment due to the increased missed speaker rate.

For the VAD task, we selected pyannote's VAD model, which is part of the open-source pyannote.audio library [5]. Pyannote.audio offers pre-trained models and pipelines for various speech processing tasks, including VAD, speaker change detection, and overlapped speech detection, achieving state-of-the-art performance across multiple datasets. Its user-friendly Python API and efficient integration capabilities made it a suitable choice

for our post-processing needs.

Model	% MS	% FA	% SC	**% DER (8 spk)**
TS-VAD+ (w/o VAD postprocessing)	**10.24**	6.92	3.44	20.60
TS-VAD+ (with VAD postprocessing)	11.68	**4.17**	**3.26**	**19.11**

Table 4.5: DER score breakdown with and without VAD postprocessing

We observe an overall reduction of approximately 1.5% in the DER with the use of VAD postprocessing. Notably, the False Alarm (false positive) rate decreased by 2.75%, while the Missed Speaker (false negative) rate increased slightly by 1.44%. The speaker confusion rate remained nearly unchanged, with a minimal difference of just 0.18%.

4.3.6 Memory Module TS-VAD+ (mm-TS-VAD+)

Finally, a set of experiments were performed using the memory module ts-vad+ to compare its performance with the vanilla TS-VAD+ model. The set of experiments were performed by first pretraining on the wideband simulated data followed by finetuning on the DIHARDIII (track II) development set. Evaluation was done on the DIHARDIII evaluation set.

S.No	**Model**	**Trained on**	**% DER**	**% MS**	**% FA**	**% SC**
1	TS-VAD+ [8 spk]	Wideband simulated data	34.18	9.94	17.78	6.45
2	mm-TS-VAD+ [8 spk]	Wideband simulated data	26.91	11.41	8.8	6.71
3	TS-VAD+ [8 spk]	Wideband simulated data, finetuned on DIHARDIII dev set	20.58	10.29	6.89	3.4
4	mm-TS-VAD+ [8 spk]	Wideband simulated data finetuned on DIHARDIII dev set	26.11	12.13	7.41	6.57

Table 4.6: % DER score comparison of TS-VAD+ vs mm-TS-VAD+

In 4.6, we compare the performance of two TS-VAD+ variants under different training regimes. Models 1 and 2 are pretrained solely on wideband simulated data and then directly evaluated on the DIHARDIII evaluation set. In contrast, models 3 and 4 are initially pretrained on simulated data and subsequently fine-tuned using the DIHARDIII development set prior to evaluation. Notably, the mm-TS-VAD+ model pretrained on simulated data (Model 2) achieves a substantially lower Diarization Error Rate (DER) of 26.91% compared to 34.18% for the conventional TS-VAD+ model (Model 1). This improvement suggests that the memory module effectively refines speaker embeddings

by leveraging cluster centroids to capture additional speaker characteristics, thereby enhancing the system's generalization to unseen data.
However, after fine-tuning, the performance differential is reduced: the TS-VAD+ model (Model 3) exhibits a pronounced DER reduction to 20.58% (a 13.6% improvement), whereas the mm-TS-VAD+ model (Model 4) shows only a marginal improvement, reducing DER from 26.91% to 26.11% (a 0.8% reduction). One hypothesis for this discrepancy is that the DIHARDIII development set is relatively small, potentially rendering it inadequate to effectively tune the additional parameters introduced by the memory module.

4.3.7 Data Augmentation of DIHARDIII development set

To address the limited size of the DIHARDIII development set and improve the robustness of our models, we employed a data augmentation strategy based on the smart mixing of audio files and their corresponding RTTM annotations. This approach generates new, realistic audio samples and aligned RTTMs, thereby artificially inflating the dataset without compromising the natural characteristics of overlapping speech.
Given that over half of the files in the DIHARDIII dev set contain two speakers and most instances of overlapping speech involve exactly two speakers, our augmentation procedure was designed to preserve this inherent structure. Specifically, we performed the following mixing strategies:

- **Single-to-Single Mixing:** We selectively combined single-speaker recordings with other single-speaker recordings. This ensured that the resulting augmented audio maintained clear speaker boundaries and realistic non-overlapping conditions.
- **Single-to-Two Mixing:** In addition to single-to-single mixing, we also mixed single-speaker files with recordings containing two speakers. This strategy was applied to simulate more complex scenarios while still preserving the typical two-speaker overlap condition prevalent in the original data.

For each mixing operation, the corresponding RTTM files were also combined and adjusted to reflect the new temporal alignments of speaker segments.
The results in 4.7 indicate that augmenting the DIHARDIII development set has a tangible positive impact on the performance of the mm-TS-VAD+ model. Specifically, when the mm-TS-VAD+ model is fine-tuned on the augmented dataset (Model 3), its DER improves to 22.36% compared to 26.11% when fine-tuned on the original DIHARDIII dev set (Model 2). This represents an absolute improvement of 3.75% in DER, bringing the performance of the mm-TS-VAD+ model much closer to that of the standard TS-VAD+ model, which achieves a DER of 20.58% (Model 1).

The improvement suggests that the augmented data—generated through smart mixing of audio files and corresponding RTTMs—provides additional training diversity, helping the mm-TS-VAD+ model better refine speaker embeddings via its memory module. Nevertheless, the mm-TS-VAD+ model still lags behind the TS-VAD+ model by 1.78% DER, indicating that there remains scope for further optimization.

S.No	Model	Trained on	% DER	% MS	% FA	% SC
1	TS-VAD+ [8 spk]	Wideband simulated data, finetuned on DIHARDIII dev set	20.58	10.29	6.89	3.4
2	mm-TS-VAD+ [8 spk]	Wideband simulated data finetuned on DIHARDIII dev set	26.11	12.13	7.41	6.57
3	mm-TS-VAD+ [8 spk]	Wideband simulated data, finetuned on augmented DIHARDIII dev set	22.36	10.02	6.03	6.31

Table 4.7: % DER score comparison of TS-VAD+ vs mm-TS-VAD+

4.3.8 Summary of results

To summarize our findings, 4.8 presents a comparative analysis of our model's performance against other state-of-the-art speaker diarization systems on the DIHARDIII evaluation set (track 2).

Model	Best % DER (8 spk) on DIHARDIII eval set (track 2)
VBx-HMM with OSD	21.97
EEND [33]	20.05
NTU's TS-VAD+ model	19.11
SOTA [19]	**16.76**

Table 4.8: DER score comparison of different models

Our proposed enhancements to the TS-VAD+ framework, including the integration of improved speaker embeddings, memory-aware attention mechanisms, and refined post-processing techniques, successfully reduced the DER from 22.57% to 19.11%. These results highlight the effectiveness of our approach in narrowing the performance gap while maintaining a robust and scalable diarization framework.

Chapter 5

Conclusion and Future Work

This thesis presented TS-VAD+, an advanced speaker diarization system that builds upon the TS-VAD framework by incorporating state-of-the-art speaker embedding models, transformer-based architectures, and novel memory-aware attention mechanisms. The proposed enhancements aimed to address key limitations of existing diarization methods, particularly in handling overlapping speech, improving speaker identity tracking, and enhancing robustness across diverse acoustic conditions.

Through extensive experimentation on the DIHARD III dataset, we systematically evaluated the impact of various model components. The choice of speaker embeddings was found to significantly influence diarization performance, with ECAPA-TDNN outperforming other architectures. Pretraining on wideband simulated data (16 kHz) provided better domain adaptation compared to narrowband data, yielding lower diarization error rates (DER). Further refinements contributed to a substantial performance boost, including improved clustering via VBx, voice activity detection (VAD) postprocessing, and data augmentation.

The introduction of the memory module in mm-TS-VAD+ demonstrated potential in leveraging learned speaker representations to refine diarization accuracy. However, while mm-TS-VAD+ improved performance when trained solely on simulated data, its benefits diminished after fine-tuning on the DIHARD III development set, likely due to insufficient training data for learning memory-aware representations. Data augmentation strategies mitigated this limitation to some extent, closing the performance gap between TS-VAD+ and mm-TS-VAD+.

Overall, the TS-VAD+ model exhibited strong diarization capabilities, outperforming traditional clustering-based and end-to-end neural diarization methods, particularly in multi-speaker and high-overlap conditions. The findings underscore the importance of robust speaker embeddings, high-quality pretraining data, and effective postprocessing techniques in advancing speaker diarization.

Future research could explore more sophisticated memory modules capable of dynamically adapting to unseen speakers while maintaining low computational complexity. Additionally, leveraging self-supervised learning techniques to improve speaker embeddings could enhance generalization across diverse domains. Finally, expanding training datasets through more extensive real-world augmentation strategies may further improve model robustness and adaptability.

Bibliography

[1] Adnan Azmat. "Transformer-Based TS-VAD with Pretrained Speech Encoder for Robust Speaker Diarization". MS Thesis. Nanyang Technological University, 2024.

[2] Jimmy Ba, Jamie Kiros, and Geoffrey Hinton. "Layer Normalization". In: (July 2016). DOI: 10.48550/arXiv.1607.06450.

[3] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 12449–12460. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/92d1e1eb1cd6f9fba3227870bb6d7f07-Paper.pdf.

[4] Aniket Bhatnagar. "Speaker Diarization — The Squad Way". In: *Hackernoon* (Sept. 2018). URL: https://medium.com/hackernoon/speaker-diarization-the-squad-way-2205e0accbda.

[5] 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*. 2019. arXiv: 1911.01255 [eess.AS]. URL: https://arxiv.org/abs/1911.01255.

[6] Sanyuan Chen et al. "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing". In: *IEEE Journal of Selected Topics in Signal Processing* 16.6 (2022), pp. 1505–1518. DOI: 10.1109/JSTSP.2022.3188113.

[7] Yafeng Chen, Siqi Zheng, Hui Wang, Luyao Cheng, Qian Chen, Shiliang Zhang, and Junjie Li. *ERes2NetV2: Boosting Short-Duration Speaker Verification Performance with Computational Efficiency*. 2024. arXiv: 2406.02167 [eess.AS]. URL: https://arxiv.org/abs/2406.02167.

[8] Christopher Cieri, David Graff, Owen Kimball, Dave Miller, and Kevin Walker. *Fisher English Training Part 2, Speech*. 2005. DOI: https://doi.org/10.35111/dz78-gx84.

[9] Marc Delcroix, Naohiro Tawara, Mireia Diez, Federico Landini, Anna Silnova, Atsunori Ogawa, Tomohiro Nakatani, Lukas Burget, and Shoko Araki. *Multi-Stream Extension of Variational Bayesian HMM Clustering (MS-VBx) for Combined End-to-End and Vector Clustering-based Diarization*. 2023. arXiv: 2305.13580 [eess.AS]. URL: https://arxiv.org/abs/2305.13580.

[10] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification". In: *Interspeech 2020*. interspeech$_2$020. ISCA, Oct. 2020. DOI: 10.21437/interspeech.2020-2650. URL: http://dx.doi.org/10.21437/Interspeech.2020-2650.

[11] Linhao Dong, Shuang Xu, and Bo Xu. "Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018, pp. 5884–5888. DOI: 10.1109/ICASSP.2018.8462506.

[12] Shaked Dovrat, Eliya Nachmani, and Lior Wolf. "Many-Speakers Single Channel Speech Separation with Optimal Permutation Training". In: Aug. 2021, pp. 3890–3894. DOI: 10.21437/Interspeech.2021-493.

[13] Yusuke Fujita, Naoyuki Kanda, Shota Horiguchi, Kenji Nagamatsu, and Shinji Watanabe. "End-to-End Neural Speaker Diarization with Permutation-Free Objectives". In: *Interspeech 2019*. Sept. 2019, pp. 4300–4304. DOI: 10.21437/Interspeech.2019-2899.

[14] John J. Godfrey and Edward Holliman. *Switchboard-1 Release 2*. 1993. DOI: https://doi.org/10.35111/sw3h-rw02.

[15] David Graff, Alexandra Canavan, and George Zipperlen. *Switchboard-2 Phase I*. 1998. DOI: https://doi.org/10.35111/c7th-nf28.

[16] David Graff, Kevin Walker, and Alexandra Canavan. *Switchboard-2 Phase II*. 1999. DOI: https://doi.org/10.35111/5qpg-1r82.

[17] Kyu Jeong Han and Shrikanth S. Narayanan. "Agglomerative hierarchical speaker clustering using incremental Gaussian mixture cluster modeling". In: *Interspeech*. 2008. URL: https://api.semanticscholar.org/CorpusID:27079907.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[19] Mao-Kui He, Jun Du, Qing-Feng Liu, and Chin-Hui Lee. "ANSD-MA-MSE: Adaptive Neural Speaker Diarization Using Memory-Aware Multi-Speaker Embedding". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 31 (2023), pp. 1561–1573. DOI: 10.1109/TASLP.2023.3265199.

[20] Shota Horiguchi, Yusuke Fujita, Shinji Watanabe, Yawen Xue, and Paola Garcia. "Encoder-Decoder Based Attractors for End-to-End Neural Diarization". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 30 (2022), pp. 1493–1507. ISSN: 2329-9304. DOI: 10.1109/taslp.2022.3162080. URL: http://dx.doi.org/10.1109/TASLP.2022.3162080.

[21] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. *HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units*. 2021. arXiv: 2106.07447 [cs.CL]. URL: https://arxiv.org/abs/2106.07447.

[22] 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: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017, pp. 5220–5224. DOI: 10.1109/ICASSP.2017.7953152.

[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". In: *Computer Speech & Language* 71 (2022), p. 101254. ISSN: 0885-2308. DOI: https://doi.org/10.1016/j.csl.2021.101254. URL: https://www.sciencedirect.com/science/article/pii/S0885230821000619.

[24] Qingjian Lin, Weicheng Cai, Lin Yang, Wang Junjie, Jun Zhang, and Ming Li. *DIHARD II is Still Hard: Experimental Results and Discussions from the DKU-LENOVO Team*. Nov. 2020. DOI: 10.21437/Odyssey.2020-15.

[25] Ivan Medennikov, Maxim Korenevsky, Tatiana Prisyach, Yuri Khokhlov, Mariya Korenevskaya, Ivan Sorokin, Tatiana Timofeeva, Anton Mitrofanov, Andrei Andrusenko, Ivan Podluzhny, Aleksandr Laptev, and Aleksei Romanenko. "Target-Speaker Voice Activity Detection: A Novel Approach for Multi-Speaker Diarization in a Dinner Party Scenario". In: *Interspeech 2020*. interspeech$_2$020. ISCA, Oct. 2020. DOI: 10.21437/interspeech.2020-1602. URL: http://dx.doi.org/10.21437/Interspeech.2020-1602.

[26] Thilo von Neumann, Christoph Boeddeker, Keisuke Kinoshita, Marc Delcroix, and Reinhold Haeb-Umbach. "Speeding Up Permutation Invariant Training for Source Separation". In: *Speech Communication; 14th ITG Conference*. 2021, pp. 1–5.

[27] Huazhong Ning, Ming Liu, and Hao Tang. "A spectral clustering approach to speaker diarization". In: *Interspeech 2006*. Vol. 5. Sept. 2006. DOI: 10.21437/Interspeech.2006-566.

[28] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. "Librispeech: An ASR corpus based on public domain audio books". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, pp. 5206–5210. DOI: 10.1109/ICASSP.2015.7178964.

[29] Mark Przybocki and Alvin Martin. *2001 NIST Speaker Recognition Evaluation Corpus*. 2002. DOI: https://doi.org/10.35111/6ex1-y346.

[30] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. "Stand-Alone Self-Attention in Vision Models". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/3416a75f4cea9109507cacd8e2f2aefc-Paper.pdf.

[31] Sourabh Ravindran, C. Demirogulu, and D.V. Anderson. "Speech recognition using filter-bank features". In: *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*. Vol. 2. 2003, 1900–1903 Vol.2. DOI: 10.1109/ACSSC.2003.1292312.

[32] Neville Ryant, Prachi Singh, Venkat Krishnamohan, Rajat Varma, Kenneth Church, Christopher Cieri, Jun Du, Sriram Ganapathy, and Mark Liberman. *The Third DIHARD Diarization Challenge*. 2021. arXiv: 2012.01477 [eess.AS]. URL: https://arxiv.org/abs/2012.01477.

[33] 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: Aug. 2021, pp. 3570–3574. DOI: 10.21437/Interspeech.2021-1208.

[34] David Snyder, Guoguo Chen, and Daniel Povey. *MUSAN: A Music, Speech, and Noise Corpus*. 2015. arXiv: 1510.08484 [cs.SD]. URL: https://arxiv.org/abs/1510.08484.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[36] Hui Wang, Siqi Zheng, Yafeng Chen, Luyao Cheng, and Qian Chen. *CAM++: A Fast and Efficient Network for Speaker Verification Using Context-Aware Masking*. 2023. arXiv: 2303.00332 [cs.SD]. URL: https://arxiv.org/abs/2303.00332.

[37] Min Xu, Ling-Yu Duan, Jianfei Cai, Liang-Tien Chia, Changsheng Xu, and Qi Tian. "HMM-Based Audio Keyword Generation". In: vol. 3333. Nov. 2004, pp. 566–574. ISBN: 978-3-540-23985-7. DOI: 10.1007/978-3-540-30543-9_71.

[38] Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen. "Permutation invariant training of deep models for speaker-independent multi-talker speech separation". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017, pp. 241–245. DOI: 10.1109/ICASSP.2017.7952154.

Appendix

A.1 Attention Mechanisms

Self-attention is a mechanism that enables a model to weigh the significance of different elements within a single sequence when forming a representation for each element. This approach allows the model to capture contextual relationships more effectively, which is particularly beneficial in tasks involving sequential data, such as natural language processing and time-series analysis. The most widely used implementation of self-attention is Scaled Dot-Product attention, which is also what we employed in our model.

A.1.1 Scaled Dot-Product Attention Mechanism

Given an input sequence of vectors $X = [\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n]$, the self-attention mechanism computes a set of output vectors $Z = [\vec{z}_1, \vec{z}_2, \ldots, \vec{z}_n]$ where each output vector $\vec{z}_i$ is a weighted sum of linearly transformed input vectors. The computation involves the following steps:

1. **Linear Transformations**: Each input vector $\vec{x}_i$ is projected into three distinct vectors: the query $\vec{q}_i$, the key $\vec{k}_i$, and the value $\vec{v}_i$. This is achieved using learned weight matrices W^Q, W^K, and W^V:

$$\vec{q}_i = W^Q \vec{x}_i, \quad \vec{k}_i = W^K \vec{x}_i, \quad \vec{v}_i = W^V \vec{x}_i$$

2. **Attention Scores**: The relevance of each input element $\vec{x}_j$ to $\vec{x}_i$ is determined by computing the dot product of their queries and keys, scaled by the square root of the dimensionality d_k of the key vectors:

$$\alpha_{ij} = \frac{\vec{q}_i \cdot \vec{k}_j^\top}{\sqrt{d_k}}$$

3. **Softmax Normalization**: The attention scores α_{ij} are normalized using the softmax function to obtain the attention weights a_{ij}:

$$a_{ij} = \frac{\exp(\alpha_{ij})}{\sum_{j=1}^{n} \exp(\alpha_{ij})}$$

4. **Weighted Sum**: Each output vector $\vec{z}_i$ is computed as a weighted sum of the value vectors $\vec{v}_j$, using the attention weights a_{ij}:

$$\vec{z}_i = \sum_{j=1}^{n} a_{ij}\vec{v}_j$$

A.1.2 Advantages of Self-Attention

The self-attention mechanism offers several benefits over traditional sequence processing methods:

- **Parallelization**: Unlike recurrent neural networks (RNNs), self-attention allows for parallel computation across all elements in the sequence, leading to more efficient training and inference.
- **Long-Range Dependencies**: Self-attention can capture relationships between distant elements in a sequence more effectively than convolutional or recurrent approaches, which may struggle with long-range dependencies.
- **Dynamic Contextualization**: Each element's representation is dynamically adjusted based on its relationship with all other elements in the sequence, providing rich and contextually aware embeddings.

A.1.3 Multi-Head Attention

Building upon the self-attention mechanism, multi-head attention extends its capacity by allowing the model to jointly attend to information from different representation subspaces. This approach enhances the model's ability to capture various aspects of the data, leading to more robust representations.

Mechanics of Multi-Head Attention

Multi-head attention involves multiple self-attention operations, referred to as "heads," each with its own set of projection parameters. The process can be outlined as follows:

- **Parallel Attention Heads**: The input X is projected into multiple sets of queries, keys, and values using distinct learned linear transformations:

$$Q_i = XW_{Q_i}, \quad K_i = XW_{K_i}, \quad V_i = XW_{V_i} \quad \text{for } i = 1, \ldots, h$$

  where h denotes the number of attention heads.

- **Individual Attention Computation**: Each set of Q_i, K_i, V_i undergoes the scaled dot-product attention mechanism independently:

$$\text{head}_i = \text{Attention}(Q_i, K_i, V_i)$$

- **Concatenation and Final Linear Projection**: The outputs from all heads are concatenated and projected through a linear layer:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W_O$$

Here, $W_O \in \mathbb{R}^{hd_v \times d}$ is a learnable weight matrix, and d_v is the dimensionality of the values in each head.

Advantages of Multi-Head Attention

- Diverse Representation Learning: By attending to information from different representation subspaces, multi-head attention enables the model to capture various features and patterns within the data.
- Enhanced Capacity: The use of multiple heads allows the model to focus on different positions and aspects of the sequence simultaneously, improving its ability to understand complex relationships.
- Stabilized Learning: Distributing the attention mechanism across multiple heads can lead to more stable gradients during training, facilitating more effective learning.

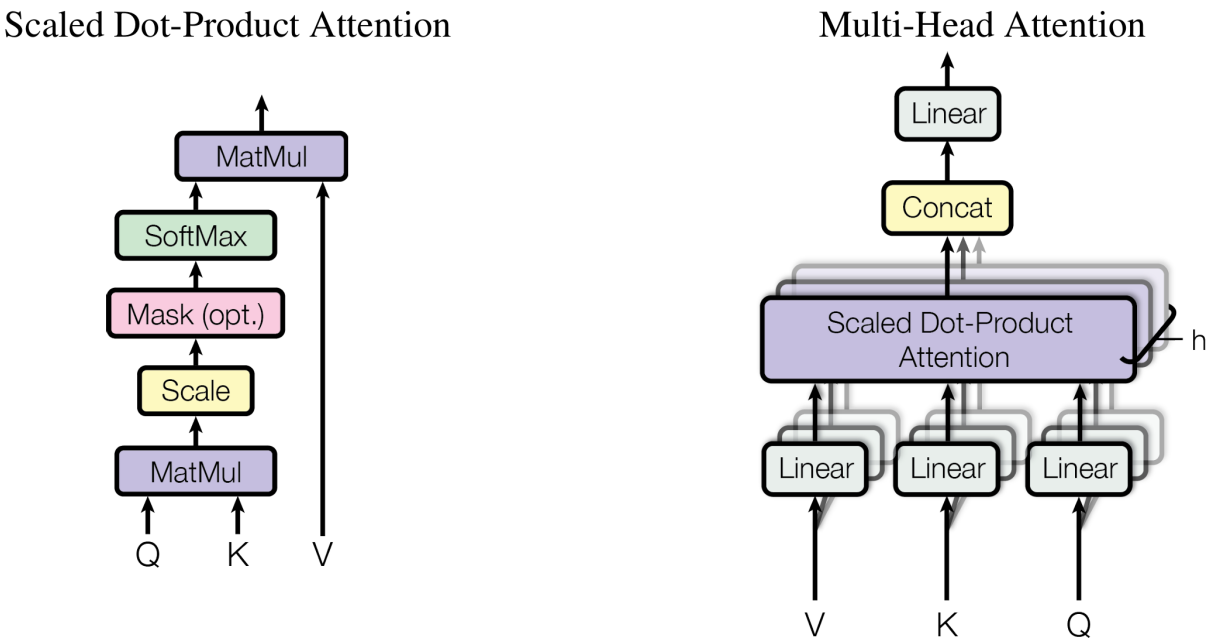

Figure 6.1: Illustration of Scaled dot product attention and Multi-head attention by Vaswani et al.[35]

A.2 Transformer Encoder Layer

The Transformer encoder layer is a fundamental building block of the Transformer architecture [35], designed to model complex dependencies in sequential data through self-attention mechanisms and position-wise feedforward networks. Unlike traditional recurrent and convolutional models, the Transformer encoder processes input sequences in parallel, significantly improving computational efficiency and capturing long-range dependencies effectively.
Each Transformer encoder layer consists of two primary sublayers: (1) a Multi-Head Self-Attention mechanism that enables the model to attend to different parts of the input sequence simultaneously, and (2) a Position-Wise Feedforward Network that refines the feature representations. To facilitate training and stabilize gradients, each sublayer is followed by residual connections [18] and layer normalization [2].

A.2.1 Multi-Head Self-Attention

Multi-head self-attention extends the standard self-attention mechanism by computing multiple attention functions in parallel, each with a different learned projection of the input. Please refer A.1

A.2.2 Position-Wise Feedforward Network

After the self-attention sublayer, the encoded representations are passed through a feedforward network, which is applied independently to each position in the sequence. The feedforward network consists of two linear transformations with a ReLU activation in between:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

where W_1, W_2, b_1, and b_2 are learnable parameters. This transformation helps in capturing complex feature interactions beyond what is achieved through self-attention alone.

A.2.3 Residual Connections and Layer Normalization

To facilitate gradient flow and stabilize training, each sublayer within the Transformer encoder employs residual connections followed by layer normalization:

$$x' = \text{LayerNorm}(x + \text{Sublayer}(x))$$

where x is the input to the sublayer, and $\text{Sublayer}(x)$ represents either the multi-head attention or feedforward network output. These components help mitigate vanishing gradient issues and improve the overall stability of deep Transformer models.

A.2.4 Applications

The Transformer encoder layer is highly effective in capturing long-range dependencies and contextual information. It along with the attention mechanism is a foundational component in models such as the Transformer [35], which has achieved state-of-the-art performance in various tasks, including machine translation, language modeling, and text summarization. Its ability to model complex dependencies and process sequences in parallel has also led to its adoption in domains like speech processing [11] and image analysis [30].

A.3 Permutation Invariant Training (PIT)

Permutation Invariant Training (PIT) is a technique designed to address the label ambiguity problem in tasks such as speaker-independent multi-talker speech separation, where the order of output sources is not predetermined [38]. This method ensures that the training process is invariant to permutations of the output labels, allowing the model to learn effectively without being affected by the arbitrary order of sources.

A.3.1 Mathematical Formulation

Consider a scenario with N mixed sources and a model that aims to separate these into N individual estimated sources. Let $S = [s_1, s_2, ..., s_N]$ represent the ground truth sources, and $\hat{S} = [\hat{s}_1, \hat{s}_2, ..., \hat{s}_N]$ denote the estimated sources produced by the model. The objective is to minimize a loss function $\mathcal{L}$ between the estimated and true sources.
In traditional training, a fixed ordering of sources is assumed, leading to a loss calculation of the form:

$$\mathcal{L}_{\text{PIT}}(\mathbf{s}, \hat{\mathbf{s}}) = \min_{\pi \in \mathcal{P}_N} \sum_{i=1}^{N} \ell\left(s_i, \hat{s}_{\pi(i)}\right)$$

where l is a sample-wise loss function, such as the mean squared error. However, in tasks like speech separation, the ordering of sources is arbitrary, and enforcing a fixed order can lead to suboptimal training.

PIT addresses this by considering all possible permutations of the source indices. Let P_n denote the set of all permutations of $\{1, 2, ..., N\}$. PIT computes the loss for each permutation and selects the minimum:

$$\mathcal{L}_{\text{PIT}}(\mathbf{s}, \hat{\mathbf{s}}) = \min_{\pi \in \mathcal{P}_N} \sum_{i=1}^{N} \ell(s_i, \hat{s}_{\pi(i)})$$

This approach ensures that the model is trained to minimize the loss for the best possible matching between the estimated and true sources, regardless of their order.

A.3.2 Computational Considerations

The number of possible permutations $|P_n|$ grows factorially with N, making the direct computation of $\mathcal{L}_{PIT}$ computationally expensive for large N. To mitigate this, approximate methods or constraints on the number of sources are often employed in practice [12, 26].

By resolving the permutation ambiguity during training, PIT enables models to learn more robust and generalizable representations in tasks where the ordering of outputs is inherently indeterminate.

A.4 Rich Transcription Time Marked File Format

The Rich Transcription Time Marked (RTTM) file format is a standard annotation format used for speaker diarization tasks. It provides time-stamped labels indicating speaker turns within an audio file. The format was originally designed for the National Institute of Standards and Technology (NIST) Rich Transcription evaluations and is widely used in diarization and speech processing research.

A.4.1 RTTM File Structure

Each line in an RTTM file represents a single speech segment and follows a specific structure with the following fields:

```
<TYPE> <FILE> <CHANNEL> <START> <DURATION> <ORT> <SPEAKER TYPE> <SPEAKER>
                                    <CONFIDENCE> <NA>
```

where:

- `<TYPE>`: Specifies the type of annotation (typically "SPEAKER" for diarization).
- `<FILE>`: The unique identifier of the audio file.
- `<CHANNEL>`: The audio channel (typically 1 for single-channel audio).
- `<START>`: Start time of the speech segment in seconds.
- `<DURATION>`: Duration of the segment in seconds.
- `<ORT>`: Orthography field (typically marked as "NA").
- `<TYPE>`: Speaker type (typically marked as "NA").
- `<SPEAKER>`: The speaker label assigned to the segment.
- `<CONFIDENCE>`: Confidence score of the annotation (often "NA" in many RTTM files).
- `<NA>`: Additional field, marked as "NA".

A.4.2 Example RTTM Entry

A typical RTTM file entry might look as follows:

```
SPEAKER example_audio 1 12.34 3.56 <NA> <NA> Speaker_1 <NA> <NA>
SPEAKER example_audio 1 16.78 2.10 <NA> <NA> Speaker_2 <NA> <NA>
```

This indicates that in the audio file `example_audio`, `Speaker_1` speaks from 12.34s to 15.90s, and `Speaker_2` speaks from 16.78s to 18.88s.

A.4.3 Usage in Speaker Diarization

RTTM files play a crucial role in speaker diarization, serving as both ground-truth labels for evaluation and as output formats for diarization systems. Metrics such as Diarization Error Rate (DER) are computed using RTTM files by comparing system-generated annotations with reference labels.

The format ensures consistency between different diarization tools and datasets, making it an essential standard in speech processing research.