WhisperDiari: A Whisper-Based Speaker Diarization Framework in Token Space Leveraging Semantic and Speaker Information for Better Text Adaptability

Yongkang Yin[1,2], **Yuexian Zou**[1,2,*]

[1]Guangdong Provincial Key Laboratory of Ultra High Definition lmmersive Media Technology, Shenzhen Graduate School, Peking University

[2]ADSPLAB, School of ECE, Peking University, Shenzhen, China

yinyongkang@stu.pku.edu.cn, zouyx@pku.edu.cn

Abstract

Speaker diarization is a fundamental task in speech processing aims to determine 'who speaks when'. When combined with ASR, it enables speaker-labeled transcription with broad practical value. Most existing methods rely on frame-level classification, but the high cost of annotating mixed-speaker audio limits the availability of large-scale, accurately labeled datasets. As a result, even state-of-the-art models struggle with imprecise speaker boundary detection and semantic segmentation errors, which degrade timestamp accuracy and downstream ASR performance. To address these challenges, we propose WhisperDiari, a unified framework for speaker diarization and ASR. We first construct LibriDiari, a dataset derived from LibriSpeech, containing 2–4 speaker mixed audio annotated with transcripts and speaker labels. WhisperDiari builds on the Whisper model, incorporating speaker adapters and Speaker Similarity Matrix Supervision to enhance speaker representation. In addition, a dedicated speaker decoder fuses speaker embeddings with contextual semantics from Whisper's decoder, enabling token-level diarization. This design effectively resolves segmentation ambiguity, aligns diarization with semantic units, and jointly models 'who speaks what and when', producing accurate, timestamped transcripts. We train the model on LibriDiari and evaluate it on both LibriDiari and the real-world AMI corpus. Experimental results demonstrate that WhisperDiari consistently outperforms state-of-the-art open-source baselines.

Introduction

Speaker diarization automatically determines "who speaks when" in multi-speaker audio (Anguera et al. 2012; Park et al. 2022). By segmenting speech and labeling speakers, it organizes audio data and improves downstream tasks like speech recognition and understanding (Kheddar, Hemis, and Himeur 2024; Cui et al. 2024). Traditional audio-based diarization methods typically assign speaker labels to short speech segments by first dividing the audio into frames. These methods are generally categorized into multi-stage and end-to-end approaches based on their processing pipelines. The multi-stage approach, one of the earliest diarization strategies, usually consists of voice activity detection (VAD) (Sharma, Rattan, and Sharma 2021), speech segmentation (Sakran et al. 2017), speaker embedding extraction (Bai and Zhang 2021), and speaker clustering. VAD detects speech regions, after which the audio is segmented into single-speaker regions. A speaker encoder then extracts embeddings from each segment, and clustering assigns speaker labels based on these embeddings. The overall performance heavily depends on the quality of each module. Improvements in VAD (Medennikov et al. 2020), segmentation (Xia et al. 2022), embedding robustness (Desplanques, Thienpondt, and Demuynck 2020; Wang et al. 2023b), and clustering accuracy (Ajmera and Wooters 2003; Han, Kim, and Narayanan 2008; Ning et al. 2006) all contribute to better results. Effective coordination and parameter tuning across modules have also led to the development of strong open-source diarization systems (Bredin et al. 2020; Chen et al. 2025). With the advancement of deep learning, end-to-end speaker diarization (EEND) methods have emerged (Fujita et al. 2019; Horiguchi et al. 2022). These methods aim to simplify the multi-stage pipeline by using a single deep neural network to directly classify each frame by speaker identity, thereby reducing error propagation across separate modules. Although EEND systems may still slightly lag behind highly optimized multi-stage systems, they show promising potential—particularly in scenarios requiring real-time processing or handling overlapping speech—thanks to their flexibility and adaptable design (Han, Lee, and Stolcke 2021; Xue et al. 2021).

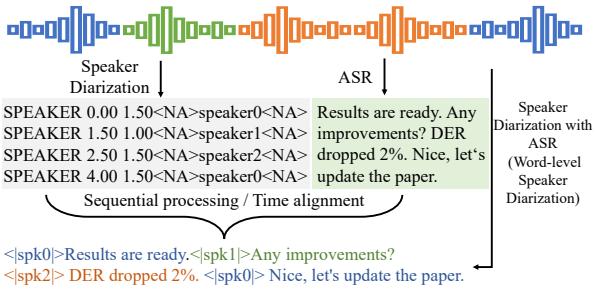

Figure 1: Combining speaker diarization and ASR or using word-Level diarization for speaker-labeled transcription

These methods typically generate segment-level speaker labels with timestamps, but this is often insufficient for applications like meeting summarization or transcription,

*indicates the corresponding author.

which require knowing 'who speaks when and what'. To address this, diarization results are often combined with automatic speech recognition (ASR) in multi-module systems as shown in Figure 1. However, since diarization and ASR are usually developed separately, differences in timing and errors like frame loss or segmentation mistakes can disrupt semantic coherence and reduce overall output quality in complex scenarios. To address this issue, recent word-level diarization methods jointly model speaker diarization and ASR by assigning speaker labels directly to transcribed words (Park et al. 2020) (Huang et al. 2023), as shown in Figure 1. This simplifies the system and improves alignment accuracy. However, these methods still have limitations. They lack deep integration of semantic and speaker information, and their weak long-range context modeling limits speaker identification performance.

To address the following challenges: 1) misalignment and semantic disruption from separate modeling of speaker diarization and ASR at different time scales; 2) limited integration of speaker and semantic information; and 3) lack of high-quality labeled data with aligned text and speaker annotations, inspired by word-level diarization methods and the Whisper model, we propose a token-level speaker diarization framework based on the Whisper architecture, along with a corresponding training and testing dataset. The main contributions are as follows:

- We propose a speaker diarization framework based on the Whisper architecture, which jointly models speaker information and semantic context;
- Our approach operates in token space while preserving timestamp outputs, and we explore a token-level evaluation metric called tDER.
- We create a multi-speaker dataset, LibriDiari based on LibriSpeech, with aligned text and speaker labels. The construction method is generalizable to other ASR datasets.
- Our method achieves competitive results on both the simulated LibriDiari and real-world AMI datasets, compared to existing open-source state-of-the-art methods.

Related Work

Whisper Model

Whisper (Radford et al. 2023) is a general-purpose speech recognition model by OpenAI. It uses a standard transformer encoder-decoder architecture (Vaswani et al. 2017) in an end-to-end pipeline. The model takes log Mel-filterbank features from raw audio as input $X \in \mathbf{R}^{T_{mel} \times D_{mel}}$, covering 30 seconds of audio. The input first passes through two 1D convolutional layers for temporal downsampling, producing compressed features. These are then processed by a multi-layer transformer encoder to extract context-aware speech representations:

$$H_{\text{conv1d}} = \text{Conv1DBlocks}(X) \in \mathbf{R}^{T_h \times D_h} \quad (1)$$

$$H_{\text{encode}} = \text{Encoder}(H_{\text{conv1d}}) \in \mathbf{R}^{T_h \times D_h} \quad (2)$$

where T_h and D_h denote the length and dimension of the hidden states after 2× downsampling. During decoding, the model generates the target token sequence autoregressively, with each token conditioned on the previously generated tokens and the encoder outputs.

$$P(y_l) = \text{Decoder}(y_{<l}, H_{\text{encode}}) \quad (3)$$

where $P(y_l)$ represents the probability of the l_{th} generated token. The training objective of the model is to minimize the standard cross-entropy loss:

$$\mathcal{L}_{\text{whisper}} = -\sum_{l=1}^{L} \log P(y_l \mid y_{<l}, H_{\text{encode}}) \quad (4)$$

where N denotes the total number of generated tokens, y_l represents the l_{th} token, $y_{<l}$ refers to all tokens generated before the l_{th} token.

The Whisper model is trained with weak supervision on 680,000 hours of multilingual audio, showing strong cross-lingual transfer ability. The training data includes labels for tasks such as ASR, speech translation, and language identification. Whisper also introduces special tokens (e.g., <|en|>, <|transcribe|>, <|timestamps|>, <|silence|>) to support various output formats and tasks. This adaptable tokenizer supports standard ASR and can be extended to tasks like speaker diarization.

Speaker Diarization Combined ASR

Joint modeling methods (or word-level speaker diarization) produce transcriptions with speaker labels within a unified framework (Park et al. 2024; Huang et al. 2024; Ma et al. 2024; Wang et al. 2024; Cornell et al. 2024). They commonly use multi-task learning or serialized output training to improve consistency between ASR and speaker labeling. However, most rely only on ASR encoder states for speaker attribution, lacking deeper fusion of semantic and speaker cues. Their complex architectures and limited open-source implementations also hinder practical adoption.

Alternatively, a common approach performs speaker diarization and ASR separately, then combines results via sequential processing or timestamp alignment. Traditional methods segment audio into single-speaker regions using diarization, then run ASR on each segment (Tranter and Reynolds 2006; Ben-Harush et al. 2012; Watanabe et al. 2020). While this reduces speaker overlap issues, it depends heavily on accurate segmentation. Recently, ASR models like Whisper provide precise timestamps, enabling more flexible strategies that align diarization and ASR outputs by timestamp for speaker labeling (Bai and Zhang 2021). Supported by tools such as pyannote.audio and Whisper, this pipeline offers a robust and reproducible solution widely used in research and practice.

Proposed Method

Data Simulation For LibriDiari

To enable end-to-end speaker diarization that jointly models semantic and speaker information using deep neural networks, triplet data containing audio, transcriptions, and

Algorithm 1: Mixture Speech Generation Algorithm

Input: M: Speech metadata
n: Number of speakers
C: Number of mixtures
Output: Speech mixtures $x = \{x_1, \ldots, x_C\}$

Initialize $count \leftarrow 0$, $x \leftarrow \{\}$
while $count < C$ **do**
  Sample $S = \{s_1, \ldots, s_n\}$ from M
  $U \leftarrow \{\}$, $R \leftarrow 30s$
  for $s_i \in S$ **do**
    $u_i \leftarrow \text{Sample}(s_i, R)$
    $U \leftarrow U \cup \{u_i\}$, $R \leftarrow R - length(u_i)$
  end for
  if $\sum_{u \in U} length(u) \leq 30s$ **then**
    while true **do**
      $u \leftarrow \text{Sample}(S, R)$
      if no such u exists **then**
        break
      end if
      $U \leftarrow U \cup \{u\}$, $R \leftarrow R - length(u)$
    end while
    $U \leftarrow \text{NormalizeLoudness}(U)$
    $x_{count} \leftarrow \text{Mix}(U)$
    $x \leftarrow x \cup \{x_{count}\}$, $count \leftarrow count + 1$
  end if
end while
return x

speaker labels is required. However, such datasets are limited. Additionally, due to Whisper's architecture and positional encoding design, the model can only process audio up to 30 seconds long. To address this, inspired by the construction method of LibriMix (Cosentino et al. 2020), we construct a dataset called LibriDiari, based on clean speech segments from LibriSpeech (Panayotov et al. 2015). Multi-speaker audio is synthesized by combining non-overlapping utterances from different speakers. Specifically, we randomly select N speakers and sample one utterance from each. If the total duration remains under 30 seconds, more utterances from the same speakers are added until the limit is nearly reached. The detailed procedure is shown in Algorithm 1

Using the same approach, corresponding noise data can be generated based on the WHAM! (Wichern et al. 2019). It is optional to add noise and perform loudness normalization of the speech segments. In addition to transcriptions and speaker IDs, the data annotations also include the duration of each utterance, as well as the silent intervals between segments, computed using a VAD model (Team 2025) and inter-utterance silence detection algorithm.

Proposed WhisperDiari Model

The proposed model consists of four main components: an acoustic encoder, a semantic decoder, a speaker decoder, and a speaker adapter. It builds on a pretrained Whisper encoder that extracts two types of features: semantic features for transcription and speaker features for identification. Since

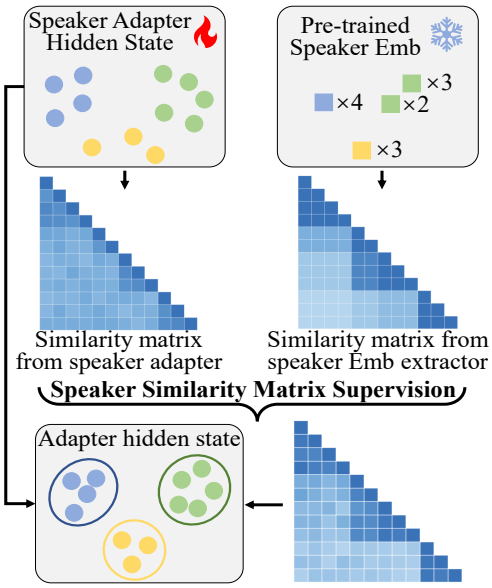

Figure 2: Speaker Similarity Matrix Supervision illustration. Enhances speaker representation robustness by leveraging pre-trained speaker embeddings.

intermediate layers of speech encoders anc capture speaker features (Zhang et al. 2024), the model uses hidden states from these layers as speaker representations, while semantic features are taken from the encoder's final layer. The semantic decoder, following Whisper's original design, autoregressively generates tokens by combining semantic features with previously generated tokens. Simultaneously, the speaker decoder uses the full sequence of semantic tokens together with speaker representations to predict speaker labels for each token, enabling diarization directly in the token space.

Speaker Adapter The original Whisper encoder processes input audio and produces representations containing semantic information. Previous studies have shown that the hidden states from intermediate layers of speech encoders retain speaker-related features. Based on this insight, the intermediate hidden states of the pretrained Whisper encoder are selected as the initial speaker representations. We introduce a trainable Speaker Adapter module to improve the distinctiveness of these representations. This module consists of a normalization layer and a bottleneck structure composed of two linear projection layers. The Speaker Adapter takes the intermediate hidden states as input and produces the final speaker representation as follows:

$$H_{\mathbf{spk}} = W_{\uparrow}\left(\mathbf{ReLU}\left(W_{\downarrow}\left(\text{LN}(H_{\mathbf{enc-mid}})\right)\right)\right) \in \mathbf{R}^{T_h \times D_h} \tag{5}$$

where the $H_{\mathbf{spk}}$ are then used as the final speaker representations for the subsequent Speaker Decoder.

Speaker Similarity Matrix Supervision Although the pretrained Whisper encoder captures some speaker information, it is not explicitly optimized for speaker discrimination, resulting in limited discriminability and robustness of

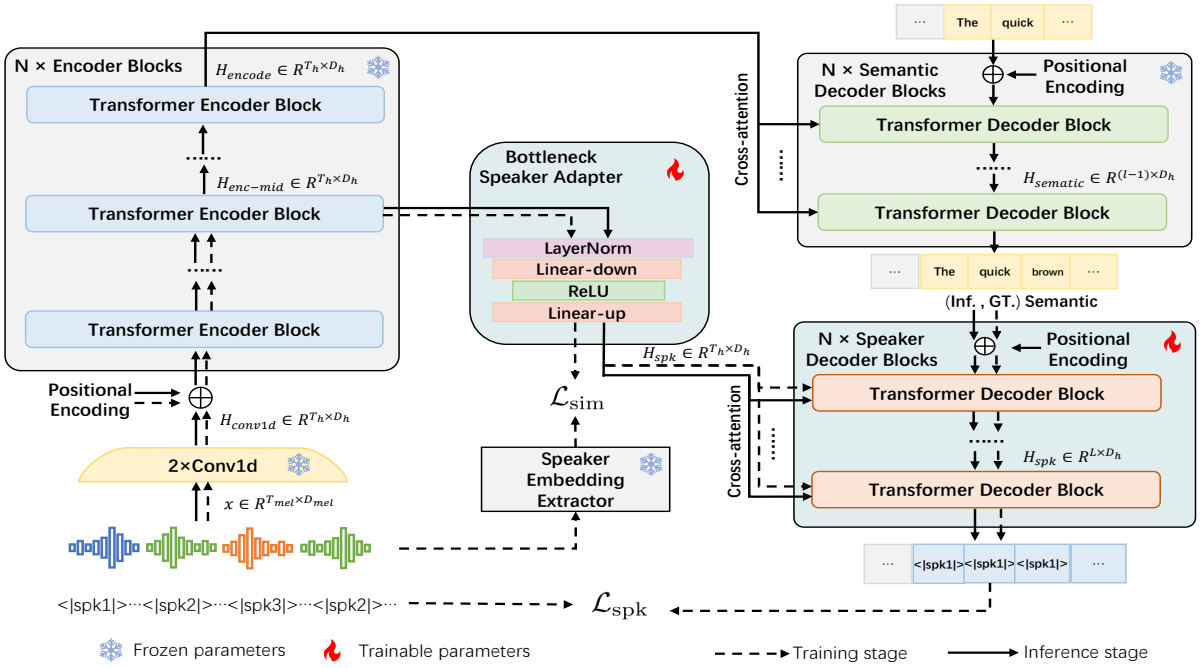

Figure 3: WhisperDiari: Extended from Whisper with a speaker adapter and decoder to enable speaker diarization in token space, using speaker and semantic information.

extracted features. Additionally, frame-level speaker embeddings often suffer from temporal instability, which can degrade downstream decoding. To overcome these issues, we propose Speaker Similarity Matrix Supervision. Leveraging a pretrained speaker embedding model as a reference, our method guides the learning of more discriminative speaker representations. Due to differences in encoding spaces and feature distributions, direct alignment is challenging, so we focus on supervising similarity relationships among embeddings to enhance discriminability while preserving the original feature distribution.

Specifically, we use the pretrained speaker verification model from the open-source WeSpeaker toolkit to extract reference speaker embeddings. Since embeddings from longer speech segments are generally more robust, we leverage the timestamp information of each continuous segment in the LibriDiari dataset. The pretrained model extracts speaker embeddings for each segment, forming $F_{\mathbf{spk}} \in \mathbf{R}^{N \times D_{\mathbf{spk}}}$, where N is the number of concatenated segments per sample. Each embedding is then temporally expanded according to its segment duration to produce the full sequence $F^{'}\mathbf{spk} \in \mathbf{R}^{T_h \times D\mathbf{spk}}$. The similarity matrices of the supervision features $F^{'}_{\mathbf{spk}}$ and the speaker adapter–adjusted intermediate hidden states $H_{\mathbf{spk}}$ from the encoder are respectively computed by calculating the pairwise similarities between frame-wise and normalized embeddings. Then two similarity matrices, M' and M, are obtained.

$$M' = \mathbf{cos}(f'_i, f'_j), \forall i, j \in [1, T_h],\ f'_i, f'_j \in \mathbf{norm}(F'_{\mathbf{spk}}) \tag{6}$$

$$M = \mathbf{cos}(h_i, h_j), \forall i, j \in [1, T_h],\ h_i, h_j \in \mathbf{norm}(H_{\mathbf{spk}}) \tag{7}$$

Finally, the mean squared error (MSE) loss between the elements of the two similarity matrices is calculated and used as the training objective.

$$\mathcal{L}_{\text{sim}} = \text{MSE}(M', M) = \frac{1}{|\mathcal{M}|} \sum_{i,j} \left(M_{i,j} - M'_{i,j}\right)^2 \cdot \mathcal{M}_{i,j} \tag{8}$$

where $\mathcal{M}$ is the mask matrix that excludes silent parts from contributing to the loss calculation.

Speaker Decoder The another core innovation of the model is the introduction of a speaker decoder on top of the original Whisper encoder-decoder architecture. This module further processes the semantic tokens generated by the semantic decoder and predicts a speaker identity label for each token. In this way, speaker diarization is performed directly in the token space, enabling fine-grained speaker tracking. Each block of the speaker decoder follows a standard transformer decoder structure, consisting of self-attention and cross-attention mechanisms. The inputs to the speaker decoder consist of the speaker representation $H_{\mathbf{spk}}$, obtained from intermediate encoder hidden states and adapted via a speaker adapter, and the complete sequence of semantic tokens $Y = \{y_1, y_2, \ldots, y_L\}$ generated by the semantic decoder:

$$y_l = \text{Logits}(\text{SemanticDec}(y_{<l}, H_{\text{enc}})) \tag{9}$$

where $H_{\text{enc}} \in \mathbf{R}^{T_h \times D_h}$ denotes the semantic representations extracted from the final layer of the encoder. The output of the speaker decoder is $Y_{\mathbf{spk}} = \{y_1^{\text{spk}}, y_2^{\text{spk}}, \ldots, y_L^{\text{spk}}\}$,

which is generated through parallel decoding, where the speaker identity for each semantic token is predicted simultaneously:

$$Y_{\mathbf{spk}} = \mathrm{Logits}(\mathrm{SpeakerDec}(Y, H_{\mathrm{spk}})) \in \mathbf{R}^{L} \tag{10}$$

where, y_l^{spk} represents the speaker identity sequence corresponding to each semantic token. To achieve this, the original tokenizer is extended with additional speaker-related special tokens, such as <|spk1|>,<|spk2|> ..., enabling token-level speaker labeling and thus realizing speaker diarization in the token space.

Optimization Targets The overall framework involves two optimization objectives. The first is the similarity-based supervision loss $\mathcal{L}_{\mathrm{sim}}$, derived from Speaker Similarity Matrix Supervision, which aims to enhance the discriminability of speaker representations. The second objective is a standard cross-entropy loss $\mathcal{L}_{\mathrm{spk}}$, which follows the loss formulation commonly used in Whisper and ASR systems, and is computed between the predicted speaker tokens and the ground-truth tokens as follows:

$$\mathcal{L}_{\mathrm{spk}} = -\sum_{l=1}^{L} \log P(y_l^{\mathrm{spk}} \mid Y, H_{\mathrm{spk}}) \tag{11}$$

The total loss is a weighted sum of the two losses:

$$\mathcal{L} = \mathcal{L}_{\mathrm{spk}} + \alpha \mathcal{L}_{\mathrm{sim}} \tag{12}$$

Experimental Setups

Datasets

LibriDiari LibriDiari is a simulated multi-speaker dataset derived from LibriSpeech's train-100h, train-360h, dev, and test subsets. It contains 2, 3, and 4 speakers speech samples, along with mixed multi-speaker training and testing data generated by randomly combining these samples. Each audio clip is 30 seconds long. Annotations include audio IDs, transcriptions, speaker labels, speech segment durations, and silence intervals. Speaker embeddings extracted from a pre-trained model are also provided to aid speaker differentiation.

Speaker Num.	train100h	train360h	dev	test
2 / 3 / 4 / n	12,000	44,000	3,000	3,000

Table 1: Number of samples in LibriDiari dataset

AMI Meeting Corpus The AMI Meeting Corpus (Kraaij et al. 2005) is a widely used multi-speaker conversational dataset comprising over 100 hours of meeting recordings, including both natural and scenario-driven sessions. Each simulated meeting typically features four participants with distinct roles in a collaborative project. The dataset offers multi-channel audio, manual transcriptions, speaker identity annotations, and voice activity labels. To facilitate processing and enhance speaker separation accuracy, we minimize overlapping speech during preprocessing while preserving the original data structure.

Setups

Model Setups The model takes a 30-second audio clip sampled at 16 kHz as input. The audio is first processed with a 25ms window and 10ms hop size, producing 3,000 frames of 80-dimensional log Mel-filterbank features. These features are then passed through two convolutional layers with an overall downsampling factor of 2×, resulting in encoder inputs with a feature dimension of 1024. Speaker labels are assigned at the token level. To ensure alignment with semantic tokens, each speaker label is duplicated based on the number of semantic tokens generated by the tokenizer for each speech segment.

The overall model is built on the pre-trained Whisper-medium architecture. Both the encoder and the two decoders use a hidden state dimension of 1024, with 24 layers and 16 attention heads. The speaker decoder is initialized from the Whisper-medium decoder and fully fine-tuned. The speaker adapter consists of two linear layers with an intermediate hidden size of 512. For speaker embedding supervision, a pre-trained CAM++ model from the open-source Wespeaker toolkit (Wang et al. 2023a) is used, generating 256-dimensional speaker embeddings. The model is trained using the Adam optimizer (Kingma and Ba 2014) with a learning rate of 10^{-4}, and the loss weight for the overall training objective is set to 0.5.

Baselines We select several widely used open-source speaker diarization methods trained on large-scale datasets as baselines. These include multi-stage approaches such as Pyannote-audio3.1, 3D-Speaker Diarization, and DiariZen (Han et al. 2024), which leverages the self-supervised WavLM model (Chen et al. 2022), as well as the end-to-end method DiaPer (Landini et al. 2023). To ensure a fair comparison and evaluate the compatibility of each diarization method with ASR, we use Whisper-medium as a consistent ASR backend across all experiments.

Metrics

Diarization Error Rate Diarization Error Rate (DER) measures the overall proportion of errors in speaker segmentation and reflects the accuracy of a speaker diarization system. It is defined as:

$$DER = \frac{T_{miss} + T_{fa} + T_{conf}}{T_{total}} \tag{13}$$

where T_{miss}, T_{fa}, and T_{conf} denote the durations of missed speech, false alarms, and speaker confusion, respectively, and T_{total} is the total reference speech duration.

Word Error Rate Word Error Rate(WER) evaluates the difference between the recognized text and the reference transcript, and is used to assess the impact of diarization on ASR performance.

$$WER = \frac{S + D + I}{N} \tag{14}$$

where S, D, and I denote the numbers of substitution, deletion, and insertion errors, respectively, and N is the total number of reference words.

Diarization Model	Diarization Style	2spk-LibriDiari		3spk-LibriDiari		4spk-LibriDiari		nspk-LibriDiari	
		DER↓	tDER↓	DER↓	tDER↓	DER↓	tDER↓	DER↓	tDER↓
Pyannote-audio	Multi-Stage	10.8	15.1	11.5	18.1	14.3	14.7	11.9	16.7
3D-Speaker	Multi-Stage	**7.7**	**6.8**	**9.5**	11.2	**10.8**	14.0	**10.2**	12.4
DiaZen	Multi-Stage	13.7	16.2	17.1	20.8	23.3	28.1	17.9	25.1
DiariPer	End-to-End	15.8	22.9	15.7	23.1	12.9	28.2	14.7	26.9
WhisperDiari	End-to-End	12.7	9.9	10.3	8.6	11.3	8.7	11.9	**8.9**
–set speaker #		11.7	7.3	11.9	**7.1**	12.2	**7.2**	-	-

Table 2: Comparison of tDER and DER on the LibriDiari dataset. WhisperDiari is trained on the n-speaker set without specifying the number of speakers, while '–set speaker #' indicates training and evaluation on subsets with fixed speaker counts. For baseline methods, tDER is computed by aligning their diarization outputs with Whisper's token-level timestamps. As all methods share the same ASR module, their WERs are identical and match the WhisperDiari results in Table 3.

Diarization Model	Inference Time (·/sample)	2spk-LibriDiari		3spk-LibriDiari		4spk-LibriDiari		nspk-LibriDiari	
		WER↓	tDER ↓	WER↓	tDER↓	WER↓	tDER↓	WER↓	tDER↓
Pyannote-audio	6.3s	6.3	12.4	5.6	15.5	6.0	15.4	5.7	15.9
3D-Speaker	4.5s	5.8	8.9	5.2	10.9	6.0	10.6	5.5	12.2
DiaZen	4.6s	13.5	31.8	13.6	32.2	17.4	45.4	14.8	34.4
DiariPer	5.2s	23.5	48.6	17.4	35.4	23.6	42.2	23.8	42.4
WhisperDiari	**4.3s**	5.6	9.9	4.8	8.6	4.7	8.7	**5.0**	**8.9**
–set speaker #		**5.6**	**7.3**	**4.8**	**7.1**	**4.7**	**7.2**	-	-

Table 3: Comparison of tDER and WER on the LibriDiari dataset. Both WhisperDiari and '–set speaker #' are described in Table 2. The Baseline performs speaker diarization followed by ASR aligned to diarized timestamps to compute speaker-attributed tokens for tDER. As diarization methods match those in Table 2, DER results are omitted.

Token-level Diarization Error Rate Token-level Diarization Error Rate (tDER) is a proposed metric for evaluating speaker label accuracy at the token level. Inspired by Word Error Rate (WER), it compares predicted and reference token sequences based on speaker labels, considering substitution, deletion, and insertion errors.

$$tDER = \frac{S_{spk} + D_{spk} + I_{spk}}{N_{spk}} \tag{15}$$

where S_{spk}, D_{spk}, and I_{spk} are the numbers of speaker token substitutions, deletions, and insertions, respectively, and N_{spk} is the number of reference tokens.

Experimental Results

Results on LibriDiari

We evaluate WhisperDiari on the LibriDiari dataset using both the standard Diarization Error Rate (DER) and the proposed token-level Diarization Error Rate (tDER), and the results remain highly consistent across multiple experiments. To ensure a fair comparison, all baseline methods perform speaker diarization and ASR separately, followed by timestamp alignment to assign speaker labels to semantic tokens. During inference, the same ASR decoding strategy is independently applied to each sample, resulting in identical Word Error Rate (WER) across datasets (see Table 3). As shown in Table 2, WhisperDiari outperforms DiariZen and DiaPer on DER, and achieves performance comparable to Pyannote-audio 3.1 and the state-of-the-art 3D-Speaker system. Notably, WhisperDiari's DER is computed from token-level timestamps generated by the decoder, which are less temporally precise than frame-level annotations used by baselines. Despite this limitation, WhisperDiari achieves significantly lower tDER, demonstrating its superior token-level speaker diarization accuracy. This improvement is attributed to direct speaker label prediction in token space and the joint modeling of diarization and ASR, which enhances temporal alignment across different granularities. Moreover, when the number of speakers is known in advance, WhisperDiari shows more stable and accurate performance under the tDER metric, highlighting its robustness in controlled scenarios.

We further evaluate WhisperDiari on LibriDiari to show the advantage of joint modeling of speaker diarization and ASR for preserving semantic coherence. The baseline uses a typical pipeline: diarization to segment and label speakers, then Whisper ASR on each segment. Results are in Table 3. Because the diarization process remains unchanged, the DER values are consistent with those reported in Table 2, avoiding duplicate statistics. Results show WhisperDiari outperforms the pipeline in both tDER and WER. The pipeline suffers from inaccurate segmentation and semantic misalignment, harming transcription coherence and raising WER. In contrast, WhisperDiari jointly optimizes speaker labels and text at the token level, reducing fragmentation and

improving accuracy and coherence.We also measured the inference time per sample for each method. Compared to the baselines, our approach achieves the fastest inference speed.

Diarization Model	Time-Align		Sequential	
	DER↓	tDER↓	WER↓	tDER↓
Pyannote-audio	22.3	25.3	18.1	29.9
3D-Speaker	18.9	21.4	19.9	27.4
DiaZen	**18.4**	19.8	19.6	26.4
DiariPer	27.7	31.2	27.6	31.0
WhisperDiari	22.7	**19.5**	**16.9**	**19.5**

Table 4: Results on AMI Corpus. Following the same evaluation protocol as on LibriDiari, we report DER and tDER computed via timestamp alignment of speaker diarization and ASR outputs, alongside WER and tDER for the sequential pipeline method.

Results on AMI Copus

We further evaluated WhisperDiari on the real-world AMI meeting corpus as shown in Figure 4. To minimize the impact of overlapping speech on recognition, we applied appropriate audio preprocessing. Experiments followed the same setup as before, comparing WhisperDiari with baselines that combine frame-level diarization and Whisper via timestamp alignment, as well as a sequential pipeline performing speaker segmentation followed by ASR on each segment.

Results show that WhisperDiari remains robust in complex real-world conditions. While its DER is competitive with state-of-the-art open-source methods, it significantly outperforms all baselines on the token-level tDER metric. Compared to the sequential pipeline, WhisperDiari also achieves lower tDER and WER, demonstrating improved semantic coherence and speaker consistency. These results confirm that joint modeling effectively addresses challenges like overlapping speech and segmentation errors, yielding more accurate and coherent transcriptions in multi-speaker conversations.

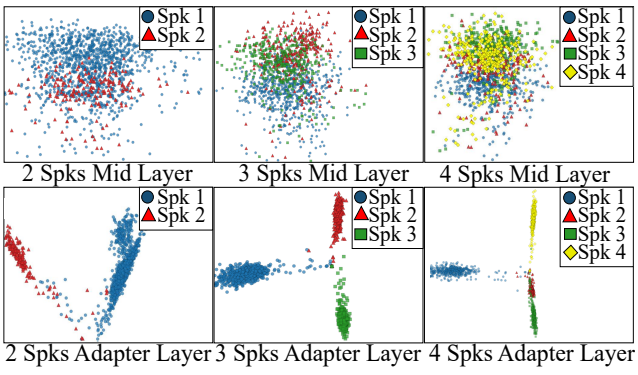

Figure 4: Visualization of speaker representations with and without adapters.Intermediate layer outputs (top) and adapter outputs (bottom) for 2/3/4 speakers (left to right).

Ablation Study

In the ablation study, we examined the effects of the Speaker Adapter and Speaker Similarity Matrix Supervision ($\mathcal{L}_{\text{sim}}$) on model performance, as well as how varying the number of similarity comparison steps influences effectiveness. Results show that both the Speaker Adapter and the cross-entropy loss substantially enhance performance and aid optimization. However, increasing the number of comparison steps in the similarity module gradually reduces the positive impact of the Speaker Similarity Matrix Supervision, further confirming its effectiveness.

In addition, we perform PCA-based (Maćkiewicz and Ratajczak 1993) dimensionality reduction to 2D and visualize both the hidden states from the encoder's intermediate layer and the adapter-adjusted representations. As shown in Figure 4, the speaker characteristics become significantly more distinguishable.

Model	nspk-LibriDiari	
	DER↓	tDER↓
WhisperDiari	**11.9**	**8.9**
$-\mathcal{L}_{\text{sim}-\text{step}} = 3$	12.6	9.4
$-\mathcal{L}_{\text{sim}-\text{step}} = 5$	13.0	10.2
–No $\mathcal{L}_{\text{sim}}$	13.1	10.5
–No Adapter	19.3	15.4

Table 5: Ablation results on the impact of the speaker adapter and similarity supervision ($\mathcal{L}_{\text{sim}}$) on model performance.

Conclusion

This paper proposes a novel token-level speaker diarization method, WhisperDiari, based on the Whisper architecture. The method not only generates speaker labels alongside the transcribed text but also integrates speaker representations with semantic contextual information at the token level. Compared to traditional frame-level or word-level diarization approaches, WhisperDiari achieves a more compact architecture and better semantic consistency. In addition, we construct a simulated multi-speaker dataset, LibriDiari, which includes fully aligned text and speaker label annotations and can serve as a standard benchmark for training and evaluating joint modeling approaches to speaker diarization. The data construction methodology can also be extended to other commonly used ASR datasets. Experimental results on both the simulated LibriDiari dataset and the real-world meeting dataset AMI Corpus demonstrate that WhisperDiari delivers competitive performance across key metrics such as DER, tDER, and WER when compared to typical combinations of speaker diarization and ASR systems. Notably, it shows clear advantages in maintaining semantic coherence and speaker label consistency, further validating the effectiveness and practical value of joint modeling in complex multi-speaker speech scenarios.

Acknowledgments

This work is supported by Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology (Grant No. 2024B1212010006).

References

Ajmera, J.; and Wooters, C. 2003. A robust speaker clustering algorithm. In *2003 IEEE Workshop on automatic speech recognition and understanding (IEEE Cat. No. 03EX721)*, 411–416. IEEE.

Anguera, X.; Bozonnet, S.; Evans, N.; Fredouille, C.; Friedland, G.; and Vinyals, O. 2012. Speaker diarization: A review of recent research. *IEEE Transactions on audio, speech, and language processing*, 20(2): 356–370.

Bai, Z.; and Zhang, X.-L. 2021. Speaker recognition based on deep learning: An overview. *Neural Networks*, 140: 65–99.

Ben-Harush, O.; Ben-Harush, O.; Lapidot, I.; and Guterman, H. 2012. Initialization of iterative-based speaker diarization systems for telephone conversations. *IEEE transactions on audio, speech, and language processing*, 20(2): 414–425.

Bredin, H.; Yin, R.; Coria, J. M.; Gelly, G.; Korshunov, P.; Lavechin, M.; Fustes, D.; Titeux, H.; Bouaziz, W.; and Gill, M.-P. 2020. Pyannote. audio: neural building blocks for speaker diarization. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7124–7128. IEEE.

Chen, S.; Wang, C.; Chen, Z.; Wu, Y.; Liu, S.; Chen, Z.; Li, J.; Kanda, N.; Yoshioka, T.; Xiao, X.; et al. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6): 1505–1518.

Chen, Y.; Zheng, S.; Wang, H.; Cheng, L.; Zhu, T.; Huang, R.; Deng, C.; Chen, Q.; Zhang, S.; Wang, W.; et al. 2025. 3D-Speaker-Toolkit: An Open-Source Toolkit for Multimodal Speaker Verification and Diarization. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Cornell, S.; Jung, J.-w.; Watanabe, S.; and Squartini, S. 2024. One model to rule them all? towards end-to-end joint speaker diarization and speech recognition. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 11856–11860. IEEE.

Cosentino, J.; Pariente, M.; Cornell, S.; Deleforge, A.; and Vincent, E. 2020. Librimix: An open-source dataset for generalizable speech separation. *arXiv preprint arXiv:2005.11262*.

Cui, W.; Yu, D.; Jiao, X.; Meng, Z.; Zhang, G.; Wang, Q.; Guo, Y.; and King, I. 2024. Recent advances in speech language models: A survey. *arXiv preprint arXiv:2410.03751*.

Desplanques, B.; Thienpondt, J.; and Demuynck, K. 2020. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *arXiv preprint arXiv:2005.07143*.

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

Han, E.; Lee, C.; and Stolcke, A. 2021. BW-EDA-EEND: Streaming end-to-end neural speaker diarization for a variable number of speakers. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7193–7197. IEEE.

Han, J.; Landini, F.; Rohdin, J.; Silnova, A.; Diez, M.; and Burget, L. 2024. Leveraging Self-Supervised Learning for Speaker Diarization. *arXiv preprint arXiv:2409.09408*.

Han, K. J.; Kim, S.; and Narayanan, S. S. 2008. Strategies to improve the robustness of agglomerative hierarchical clustering under data source variation for speaker diarization. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(8): 1590–1601.

Horiguchi, S.; Fujita, Y.; Watanabe, S.; Xue, Y.; and Garcia, P. 2022. Encoder-decoder based attractors for end-to-end neural diarization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30: 1493–1507.

Huang, Y.; Wang, W.; Zhao, G.; Liao, H.; Xia, W.; and Wang, Q. 2023. Towards word-level end-to-end neural speaker diarization with auxiliary network. *arXiv preprint arXiv:2309.08489*.

Huang, Y.; Wang, W.; Zhao, G.; Liao, H.; Xia, W.; and Wang, Q. 2024. On the Success and Limitations of Auxiliary Network Based Word-Level End-to-End Neural Speaker Diarization. In *Proc. Interspeech 2024*, 32–36.

Kheddar, H.; Hemis, M.; and Himeur, Y. 2024. Automatic speech recognition using advanced deep learning approaches: A survey. *Information fusion*, 109: 102422.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kraaij, W.; Hain, T.; Lincoln, M.; and Post, W. 2005. The AMI meeting corpus. In *Proc. International Conference on Methods and Techniques in Behavioral Research*, 1–4.

Landini, F.; Diez, M.; Stafylakis, T.; and Burget, L. 2023. DiaPer: End-to-End Neural Diarization with Perceiver-Based Attractors. *arXiv preprint arXiv:2312.04324*.

Ma, F.; Tu, Y.; He, M.; Wang, R.; Niu, S.; Sun, L.; Ye, Z.; Du, J.; Pan, J.; and Lee, C.-H. 2024. A spatial long-term iterative mask estimation approach for multi-channel speaker diarization and speech recognition. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 12331–12335. IEEE.

Maćkiewicz, A.; and Ratajczak, W. 1993. Principal components analysis (PCA). *Computers & Geosciences*, 19(3): 303–342.

Medennikov, I.; Korenevsky, M.; Prisyach, T.; Khokhlov, Y.; Korenevskaya, M.; Sorokin, I.; Timofeeva, T.; Mitrofanov, A.; Andrusenko, A.; Podluzhny, I.; et al. 2020. Target-speaker voice activity detection: a novel approach for multi-speaker diarization in a dinner party scenario. *arXiv preprint arXiv:2005.07272*.

Ning, H.; Liu, M.; Tang, H.; and Huang, T. S. 2006. A spectral clustering approach to speaker diarization. In *Interspeech*.

Panayotov, V.; Chen, G.; Povey, D.; and Khudanpur, S. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 5206–5210. IEEE.

Park, T.; Medennikov, I.; Dhawan, K.; Wang, W.; Huang, H.; Koluguri, N. R.; Puvvada, K. C.; Balam, J.; and Ginsburg, B. 2024. Sortformer: Seamless integration of speaker diarization and asr by bridging timestamps and tokens. *arXiv preprint arXiv:2409.06656*.

Park, T. J.; Han, K. J.; Huang, J.; He, X.; Zhou, B.; Georgiou, P.; and Narayanan, S. 2020. Speaker diarization with lexical information. *arXiv preprint arXiv:2004.06756*.

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

Radford, A.; Kim, J. W.; Xu, T.; Brockman, G.; McLeavey, C.; and Sutskever, I. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, 28492–28518. PMLR.

Sakran, A. E.; Abdou, S. M.; Hamid, S. E.; and Rashwan, M. 2017. A review: Automatic speech segmentation. *International Journal of Computer Science and Mobile Computing*, 6(4): 308–315.

Sharma, S.; Rattan, P.; and Sharma, A. 2021. Recent developments, challenges, and future scope of voice activity detection schemes—a review. *Information and Communication Technology for Competitive Strategies (ICTCS 2020) Intelligent Strategies for ICT*, 457–464.

Team, T. 2025. TEN VAD: A Low-Latency, Lightweight and High-Performance Streaming Voice Activity Detector (VAD). https://github.com/TEN-framework/ten-vad.git.

Tranter, S. E.; and Reynolds, D. A. 2006. An overview of automatic speaker diarization systems. *IEEE Transactions on audio, speech, and language processing*, 14(5): 1557–1565.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, H.; Liang, C.; Wang, S.; Chen, Z.; Zhang, B.; Xiang, X.; Deng, Y.; and Qian, Y. 2023a. Wespeaker: A research and production oriented speaker embedding learning toolkit. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Wang, H.; Zheng, S.; Chen, Y.; Cheng, L.; and Chen, Q. 2023b. Cam++: A fast and efficient network for speaker verification using context-aware masking. *arXiv preprint arXiv:2303.00332*.

Wang, W.; Cai, D.; Cheng, M.; and Li, M. 2024. Joint Inference of Speaker Diarization and ASR with Multi-Stage Information Sharing. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 11011–11015. IEEE.

Watanabe, S.; Mandel, M.; Barker, J.; Vincent, E.; Arora, A.; Chang, X.; Khudanpur, S.; Manohar, V.; Povey, D.; Raj, D.; et al. 2020. CHiME-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings. *arXiv preprint arXiv:2004.09249*.

Wichern, G.; Antognini, J.; Flynn, M.; Zhu, L. R.; McQuinn, E.; Crow, D.; Manilow, E.; and Roux, J. L. 2019. Wham!: Extending speech separation to noisy environments. *arXiv preprint arXiv:1907.01160*.

Xia, W.; Lu, H.; Wang, Q.; Tripathi, A.; Huang, Y.; Moreno, I. L.; and Sak, H. 2022. Turn-to-diarize: Online speaker diarization constrained by transformer transducer speaker turn detection. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8077–8081. IEEE.

Xue, Y.; Horiguchi, S.; Fujita, Y.; Watanabe, S.; García, P.; and Nagamatsu, K. 2021. Online end-to-end neural diarization with speaker-tracing buffer. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, 841–848. IEEE.

Zhang, L.; Jiang, N.; Wang, Q.; Li, Y.; Lu, Q.; and Xie, L. 2024. Whisper-SV: Adapting Whisper for low-data-resource speaker verification. *Speech Communication*, 163: 103103.