USED: Universal Speaker Extraction and Diarization

Junyi Ao, Mehmet Sinan Yıldırım, Ruijie Tao, Meng Ge, Shuai Wang, *Member, IEEE*, Yanmin Qian, *Senior Member, IEEE*, Haizhou Li, *Fellow, IEEE*

***Abstract*—Speaker extraction and diarization are two enabling techniques for real-world speech applications. Speaker extraction aims to extract a target speaker's voice from a speech mixture, while speaker diarization demarcates speech segments by speaker, annotating 'who spoke when'. Previous studies have typically treated the two tasks independently. In practical applications, it is more meaningful to have knowledge about 'who spoke what and when', which is captured by the two tasks. The two tasks share a similar objective of disentangling speakers. Speaker extraction operates in the frequency domain, whereas diarization is in the temporal domain. It is logical to believe that speaker activities obtained from speaker diarization can benefit speaker extraction, while the extracted speech offers more accurate speaker activity detection than the speech mixture. In this paper, we propose a unified model called Universal Speaker Extraction and Diarization (USED) to address output inconsistency and scenario mismatch issues. It is designed to manage speech mixtures with varying overlap ratios and variable number of speakers. We show that the USED model significantly outperforms the competitive baselines for speaker extraction and diarization tasks on LibriMix and SparseLibriMix datasets. We further validate the diarization performance on CALLHOME, a dataset based on real recordings, and experimental results indicate that our model surpasses recently proposed approaches.**

***Index Terms*—speaker extraction, speaker diarization, multi-talker scenario, LibriMix, CALLHOME**

I. Introduction

EACH speaker has unique voice characteristics. Speaker extraction seeks to extract speech from the target speaker with reference to his/her voiceprint [1], while speaker diarization analyzes the audio signal to determine "who spoke when" [2]. As important speech processing front-ends, they are widely used in various real-world speech applications, such as speaker verification [3], [4] and speech recognition [5]–[8].

In a multi-talker scenario, the human brain processes incoming speech signals by performing speaker extraction and diarization simultaneously. Prior studies typically treated speaker extraction [9]–[11] and diarization [12]–[16] as two independent tasks. However, in practical applications, such as meeting analysis and mobile recording applications, solely relying on the results from one task is insufficient since it can only provide speaking boundaries (timestamp information) or clean speech (content information). Associating extracted speech and speaker labels offers more meaningful and interpretable results as it provides a comprehensive view of "who spoke what and when" [17]. Moreover, this association provides clean speech signals so that it can further facilitate the speaker-attributed automatic speech recognition task, which aims to answer "who spoke what" [18]–[20].

Given their shared objective of disentangling speakers from a speech mixture, we believe that speaker extraction and speaker diarization complement one another. On the one hand, speaker extraction leads to a clean speech signal, therefore enhancing the effectiveness of speaker activity detection in speaker diarization. On the other hand, the information regarding the presence or absence of the target speaker resulting from speaker diarization is helpful for speaker extraction.

However, integrating speaker extraction and speaker diarization is challenging because of several critical differences between these tasks. First, their outputs are presented in very different ways. A speaker extraction system [9]–[11] typically generates a single speaker's voice. In contrast, a speaker diarization system [12]–[14] seeks to annotate the speech activities for all speakers. Second, their application scenarios are significantly different. A speaker extraction system is usually optimized for highly overlapped speech, e.g. with a speaker overlapping ratio of near 100% [21]. Conversely, speaker diarization studies mainly handle situations with sparsely overlapped speech, as the speaker overlap ratio is around 20% in daily conversations and meetings [22], [23]. We define these two differences as the 'output inconsistency' and the 'scenario mismatch' issues, which need to be addressed when integrating them.

Most of the previous studies dealt with speaker extraction and speaker diarization either independently [9], [11], [13], [14] or jointly under some controlled conditions [24], [25], such as with a fixed number of speakers or a pre-defined overlap ratio. To date, no single-model approach has been proposed to perform speaker extraction and speaker diarization for speech mixtures with varying overlap ratios and variable number of speakers. This prompts us to look into effectively incorporating speaker extraction and speaker diarization to benefit from their interaction.

With this motivation, we propose the Universal Speaker

Junyi Ao, and Haizhou Li are with Shenzhen Research Institute of Big Data, School of Data Science, The Chinese University of Hong Kong, Shenzhen 518172, China (e-mail: junyiao1@link.cuhk.edu.cn; haizhouli@cuhk.edu.cn).

Mehmet Sinan Yıldırım, and Ruijie Tao are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077 (e-mail: sinan@u.nus.edu; ruijie@nus.edu.sg).

Meng Ge is with Saw Swee Hock School of Public Health, National University of Singapore, Singapore 117549 (e-mail: gemeng@nus.edu.sg)

Shuai Wang is with Shenzhen Research Institute of Big Data, Shenzhen 518172, China (e-mail:wangshuai@cuhk.edu.cn)

Yanmin Qian is with the Auditory Cognition and Computational Acoustics Lab, the Department of Computer Science and Engineering and the MoE Key Laboratory of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: yanminqian@sjtu.edu.cn).

This article has been accepted for publication in IEEE/ACM Transactions on Audio, Speech and Language Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2024.3511268

Extraction and Diarization model (USED) to address output inconsistency and scenario mismatch. The term 'universal' is defined by two aspects: (a) the ability to perform for a variable number of speakers, and (b) the ability to process speech mixtures with an arbitrary overlap ratio.

To this end, we design the model by considering two aspects: data ingress and data egress. From a data ingress perspective, we design a novel embedding assignment module driven by speech references to mitigate output inconsistency naturally. This module yields two advantages. Firstly, it enables the generation of outputs for a variable number of speakers according to the speech references, thereby ensuring consistency across tasks. Secondly, it helps the model align speaker extraction and speaker diarization outputs to avoid the output permutation problem [26]. From a data egress perspective, we extend the scope of overlap ratio by a multi-task interaction module to alleviate the scenario mismatch between speaker extraction and speaker diarization. Specifically, the multi-task interaction module leverages speaker diarization results to mitigate background noise of the extracted speech and enhance the optimization of scenario-aware differentiated (SAD) loss [27] within our USED. Simultaneously, speaker extraction produces a cleaner speech signal to assist in the prediction of speaker diarization, particularly for the overlap parts.

The contributions of this paper can be summarized as follows:

- We propose a universal solution, i.e. USED, that performs the speaker diarization and extraction jointly. The solution deals with speech mixtures with an arbitrary number of speakers and diverse overlap ratios.
- We design an embedding assignment module to ensure consistency in the number and the order of model outputs. This aids in supporting the generation of results for a variable number of speakers and enhances the model's robustness.
- We design a novel multi-task interaction module with a scenario-aware differentiated loss that allows the diarization output to control whether the target speech is silenced, thereby ensuring temporal overlap consistency between the diarization and extraction outputs.
- Our USED model outperforms competitive baseline systems of speaker extraction and speaker diarization on both the LibriMix and SparseLibriMix datasets. Additionally, we assess the diarization performance on the CALLHOME dataset, which consists of real recordings. The experimental results show that our model performs better with significantly less pre-training data.

The rest of the paper is organized as follows. Section II discusses the related work. Section III describes the proposed method. We introduce the experimental setup in Section IV. Section V presents the experimental results and analysis. Section VI concludes the study.

II. Related Work

A. Speech Separation and Speaker Extraction

Recent studies on blind speech separation have made significant progress, with the development of advanced models such as Conv-TasNet [28], Dual-Path Recurrent Neural Network (DPRNN) [29], Band-Split Recurrent Neural Network (BSRNN) [30] and efficient training algorithms such as Permutation Invariant Training (PIT) [26]. However, speech separation methods suffer from the global permutation ambiguity problem [10], which makes it hard to process long utterances. Moreover, it requires prior knowledge or estimation of the number of speakers in the speech mixture in advance, which is hard to get in real-world applications. Unlike blind speech separation that seeks to separate all speakers, speaker extraction only extracts speech for one target speaker at a time. In general, speaker extraction methods are classified into frequency-domain methods, such as SpeakerBeam [31]–[33] and VoiceFilter [34], and time-domain methods, e.g. SpEx+ [9] and X-SepFormer [11]. Time-domain approaches can naturally avoid the phase estimation problem, which exists in frequency-domain approaches [9]. Therefore, we develop the USED model based on the time-domain approach, SpEx+.

It is common that speech separation and speaker extraction systems in the literature are optimized for highly overlapped speech mixtures [9], [11], [28]. However, the overlap ratio in daily conversations and meetings is typically around 20% [22], [23], leading to a scenario mismatch between training on highly overlapped speech and evaluating on sparsely overlapped speech. To address such mismatch, speaker extraction for sparsely overlapped speech was recently studied [27], [35], [36]. It was suggested to minimize speech power when the target speaker is absent, while maximizing the signal-to-distortion ratio (SDR) or scale-invariant signal-to-distortion ratio (SI-SDR) [37] when the target speaker is present. This introduces another challenge since it is hard to balance the two scenario-dependent objectives, i.e., SI-SDR and power, during training. We will study a multi-task interaction module in the USED model to overcome this challenge.

B. Speaker Diarization

Traditional clustering-based diarization methods [38]–[42] typically comprise multiple components that are trained separately. These methods implicitly assume that each speech segment is only from one speaker, which fails when multiple people speak at the same time. Two mainstream studies are attempting to solve this problem, i.e. end-to-end neural diarization (EEND) [12], [13], and target speaker voice activity detection (TS-VAD) [14]. EEND seeks to minimize the diarization error directly with permutation-free objectives [12], while TS-VAD uses the speaker embedding from the clustering results as the reference to detect the speaking status of each person. These methods have obtained good performance and set the stage for our study.

C. Interaction Between Speech Separation and Diarization

There were studies on the interaction between speech separation and diarization [12], [43] by training speech processing models with multiple tasks. EEND-SS [25] is a joint end-to-end framework for speaker diarization and separation. It utilizes the bottleneck feature of separation as one of the inputs for diarization and is optimized using multi-task learning. This model faces challenges related to permutation and processing very long sentences. In addition, the fusion technique of EEND-SS is only applied during inference by using diarization probability directly. In TS-SEP [24], it was proposed to pre-train a model with TS-VAD objective, and subsequently finetune the model to predict the mask for separation. In other words, TS-SEP is a two-stage framework that focuses on the speech separation task. Unlike EEND-SS and TS-SEP, our USED is a universal model that can concurrently handle both tasks and address their output inconsistency and scenario mismatch issues.

Two recent works [44], [45] have explored the joint training of speech separation and speaker diarization based on unsupervised speech separation approaches. PixIT [44] employs PIT loss to integrate the speaker diarization task with MixIT [46] and draws on the *best-of-both-worlds* framework [47], [48] to stitch different segments based on the speaker diarization results. On the other hand, Neural FCASA [45] combines speaker diarization with neural FCA [49], [50], eliminating the need for a separate speaker diarization step to mask source activities. Both approaches jointly address the two tasks under the premise of unsupervised speech separation, focusing on long-form audio and optimizing the final separation results in terms of diarization error rate (DER) and word error rate (WER). In comparison, our USED model emphasizes the performance of supervised speech separation and speaker diarization across different overlap ratios and scenarios.

III. Universal Speaker Extraction and Diarization

A. Task Definition

Let x be a multi-talker speech mixture with varying overlap ratio from 0% to 100 %, which consists of the speech signals from l speakers and a noise signal n. The speech mixture can be represented as,

$$x = \sum_{i=1}^{l} c_i + n \tag{1}$$

where $l > 1$, and c_i denotes the speech signal of speaker i. Each speaker i has a corresponding speech reference, denoted as z_i, which is estimated from the speech mixture or provided as pre-enrolled information. Note that no normalization or noise reduction techniques are applied to the raw speech signals.

Given a speech mixture and speech references, the goal of the USED model is to generate separated speech signals and speaker activity information concurrently for a variable number of speakers, ranging from 1 to l. The number of speakers corresponds to the available speech references. This is motivated by scenarios where obtaining speech references for all speakers is challenging, or the target speakers are only a subset of all speakers. During inference, if there is only one target speaker, the USED model outputs a speech signal and diarization result for that speaker in a similar way to a speaker extraction model [9]. However, if there are l target speakers, the model would output multiple speech signals and a diarization results concurrently, one for each speaker, just like a speech separation model does [28].

B. System Overview

As illustrated in Fig. 1, the USED model comprises six components: a speech encoder, a speaker encoder, an embedding processing module, a separator, an extraction decoder, and a diarization decoder.

The speaker encoder converts the available speech references into speaker embeddings, that are the feature representations of the target speakers. Meanwhile, the speech encoder generates spectrum or spectrum-like feature representations from the speech mixture x.

Unlike TS-VAD [14], which generates diarization results for all speakers according to speaker embeddings and the speech mixture, we propose an embedding assignment module to enable custom control over the number of model's outputs. The embedding assignment module prepares the input embeddings with three different states, *active*, *blank*, and *residual*, based on the speaker embeddings, which are responsible for generating results accordingly. The *active* state is used as a position marker for the expected speakers. The *blank* state serves as a position marker to guide the network in outputting a silent segment. The *residual* state marks the residual unexpected speakers. The *blank* state is motivated by Connectionist Temporal Classification (CTC) [51], which integrates blank characters to manage pauses in speech recognition and gaps between characters in Optical Character Recognition (OCR). The embedding with *blank* state is designed to clearly distinguish the outputs from the expected and unexpected speakers. The reason for this design is that the output from an unexpected speaker is usually a silent segment. With the help of the embedding assignment module, the USED model naturally avoids the output inconsistency stemming from the integration of speaker extraction and diarization.

The separator then aims to separate speakers at the representation level from a speech mixture and embeddings from the embedding assignment module. Finally, diarization and extraction decoders predict the corresponding results based on the output of the separator.

The USED model differs from typical speaker extraction and speaker diarization systems [9], [12] in two key aspects. Firstly, it outputs speech signals for an arbitrary number of speakers, ranging from 1 to l, with the help of the embedding assignment module. Secondly, it seeks to effectively deal with speech mixtures with a wide range of overlapping ratios.

C. Speech Encoder

Analogous to a frequency analyzer, the speech encoder aims to generate spectrum-like frame-based embedding sequences from the speech mixture and speech references. Inspired by SpEx+ [9], our approach leverages a multi-scale speech encoder.

Our multi-scale speech encoder is structured as a combination of three one-dimensional Convolutional Neural Networks (Conv1D), each utilizing distinct kernel sizes to represent different scales. Subsequent to the convolutional layers, the activation function applied is the Rectified Linear Unit (ReLU).

The speech encoder produces spectrum-like frame-based embedding sequences X and Z_i for 3 scales as defined in

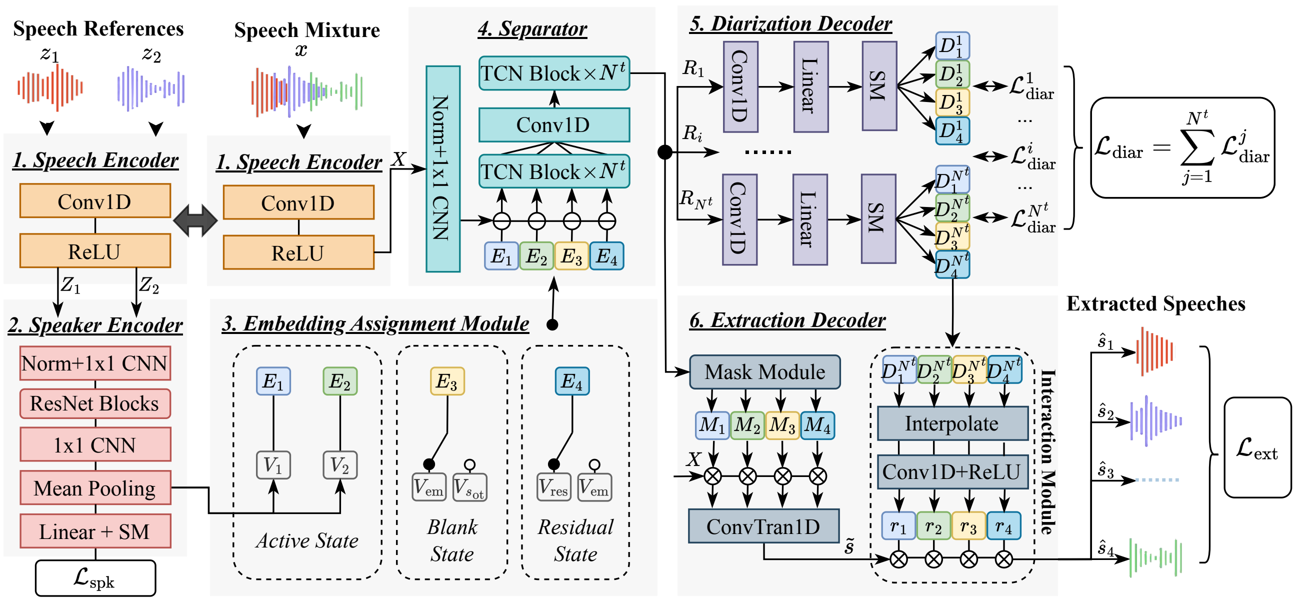
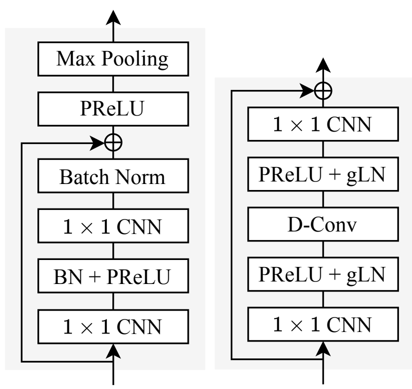


Fig. 1. The overall training and inference framework of our proposed USED model. It comprises a speech encoder, a speaker encoder, an embedding assignment module, a separator, an extraction decoder and a diarization decoder. The USED model generates both the extracted speech and speaker diarization results by leveraging the speech mixture and the speech references as input. The symbols ⊗ and ⊖ refer to element-wise multiplication and frame-wise concatenation, respectively. The TCN block and mask module are denoted by rounded rectangles, which are described in detail in Fig. 2 and introduced later. Different colours are used to distinguish network layers from different components. The *active*, *blank* and *residual* states are assigned by Algorithm 1.



(a) ResNet Block (b) TCN Layer

Fig. 2. Model structure for a ResNet block and a TCN layer. BN, PReLU, gLN and D-Conv are batch normalization, parametric ReLU, global layer normalization and dilated depth-wise separable convolution. The symbol ⊕ refers to element-wise addition.

[9] from the speech mixture x and the i-th speech reference z_i as

$$X = [X_1, X_2, X_3] = e(x) \in \mathbb{R}^{3 \times T \times C} \tag{2}$$

$$Z_i = [Z_{i1}, Z_{i2}, Z_{i3}] = e(z_i) \in \mathbb{R}^{3 \times T \times C} \tag{3}$$

where the function $e(\cdot)$ denotes the operation performed by the multi-scale speech encoder. The set $Z = \{Z_1, ..., Z_l\}$ represents the embedding sequences for l speakers. The input and output channel sizes of the three Conv1Ds are one and C, respectively. The kernel sizes of the three Conv1Ds are denoted as L_1^e, L_2^e and L_3^e, while they have a common stride of $\frac{L_1^e}{2}$. The outputs X and Z_i have a shape of $3 \times T \times C$, where T is the number of frames. For simplicity, Fig. 1 does not explicitly depict the multi-scale structure.

D. Speaker Encoder

The speaker encoder is designed to extract speaker embeddings from the corresponding enrolled speech. Specifically, we employ a speaker encoder to extract speaker embeddings, denoted as $V_1, ..., V_l$, from the embedding sequences Z. These speaker embeddings capture the distinctive characteristics of each speaker. After layer normalization (LN), we leverage a CNN layer with a kernel size of 1×1 (1×1 CNN) on the embedding sequence Z_i for speaker i. This is subsequently followed by N^r residual network (ResNet) blocks [52]. The input and output channel sizes of this 1×1 CNN are $3C$ and C, respectively. Finally, an additional 1×1 CNN, combined with a mean pooling operation, generates a speaker embedding V_i for speaker i with an embedding dimension D_{spk}. After that, the speaker encoder predicts speaker identities, denoted as $\hat{y}_i^{\mathrm{spk}}$, through a linear layer followed by a softmax (SM) activation function for speaker i.

The detailed architecture of the ResNet we used is shown in Fig. 2a. A ResNet block comprises two 1×1 CNNs and a max-pooling layer. Each 1×1 CNN within the ResNet block is accompanied by batch normalization (BN) and parametric ReLU (PReLU) for normalization and non-linear transformation. A skip connection adds the input to the output obtained after the second BN layer. Lastly, the max-pooling layer is responsible for adjusting the length of the output embedding.

We choose ResNet [52] because it has become a prominent architecture in speaker verification systems [53]–[56] and is widely adopted in speaker extraction models [9], [57]. Its key advantage lies in the residual connections, which link frame-level layers and mitigate the vanishing gradient problem, facilitating faster convergence during backpropagation [52]. These residual connections also enable deeper neural network construction, allowing ResNet-based models to outperform

Algorithm 1: Assigning States for Embeddings

Data:
k: Maximum number of speakers the model can process
$S_{\text{mix\_spk}}$: Set of speaker IDs for the speech mixture
$S_{\text{all\_spk}}$: Set of speaker IDs for the whole dataset
V_{em}: Empty embedding
V_{res}: Residual embedding
p_{sel}: Probability for selecting present speaker
p_{em}: Probability to use empty embedding as input
p_{res}: Probability to use residual embedding as input
// List of embeddings with active and blank state
$I_{active}, I_{blank} \leftarrow [\,], [\,]$;
// Number of embeddings
$count \leftarrow 0$;
/* The First Step for Active State */
for $s_{mix} \in S_{mix\_spk}$ **do**
    if s_{mix} *is a present speaker* **then**
        $p \leftarrow$ random_uniform$(0, 1)$;
        // If true, then marking embedding as *active* state
        if $p > p_{sel}$ **then**
            $E_{count} \leftarrow V_{s_{\text{mix}}}$;
            I_{active}.add(E_{count});
            $count \leftarrow count + 1$;
        end
    end
end
/* The Second Step for Blank State */
while $count \leq k$ **do**
    $p \leftarrow$ random_uniform$(0, 1)$;
    if $p > p_{em}$ **then**
        // Randomly select one speaker from the set of speakers that includes $S_{\text{all\_spk}}$ but excluding $S_{\text{mix\_spk}}$
        $s_{\text{ot}} \leftarrow$random_sample$(S_{\text{all_spk}} \setminus S_{\text{mix_spk}})$;
        $E_{count} \leftarrow V_{s_{\text{ot}}}$;
    else
        $E_{count} \leftarrow V_{\text{em}}$;
    end
    I_{blank}.add(E_{count});
    $count \leftarrow count + 1$;
end
/* The Third Step for Residual State */
$p \leftarrow$ random_uniform$(0, 1)$;
if $p > p_{res}$ *or* I_{active}*.length* $\neq$ *number of present speakers* **then**
    $E_{k+1} \leftarrow V_{\text{res}}$;
else
    $E_{k+1} \leftarrow V_{\text{em}}$;
end
Output: I_{active}, I_{blank}, E_{k+1}

regular CNNs. As a result, using ResNet as the backbone for speaker encoders leads to more robust and higher-quality speaker embeddings, making it a natural choice for this task.

E. Embedding Assignment Module

The embedding assignment module aims to prepare embeddings with different states for the separator. We set embeddings with different states: *active*, *blank*, and *residual*. The details of the embedding assignment module are shown in Algorithm 1.

1) Active State: The embedding with *active* state is designed to guide the network to generate outputs for the expected speakers. Based on the embedding with *active* state, the whole speaker extraction and diarization process is driven by the speaker encoder that takes the speech references of some present speakers as input. Specifically, to enable our model to achieve the correct output for an arbitrary number of speakers, i.e., from 1 to l, we randomly assign the *active* state to the embeddings of present speakers by a hyperparameter p_{sel} representing probability, as shown in the first step of Algorithm 1.

2) Blank State: The embeddings with *blank* state are primarily responsible for driving the network to produce silent segments for the unexpected speakers, which is critical in ensuring our network's robustness. Unlike *active* state, which requires a unique embedding from the speaker encoder, the embeddings with *blank* state alternate between two embeddings via a threshold p_{em} to guide the network: an inactive speaker embedding and a learnable empty embedding. The inactive speaker embedding is derived from a speaker who is not present in the speech mixture. This design ensures that our USED model can customize the number of outputs by inputting the learnable empty embedding during the inference stage.

3) Residual State: As an additional state, the targets corresponding to the embedding with *residual* state are the cumulative outputs for speakers that are present but are not selected at the first step. This design allows our USED model to explore the interaction among all speakers in the speech mixture. Specifically, we also employ a learnable embedding V_{res} to teach the network to learn the corresponding cumulative outputs for the residuals. When speaker embeddings of all speakers have been considered as the *active* state, the target corresponding to *residual* state should be left without any speakers. In such cases, the choice between V_{res} and V_{em} is determined through a specified threshold p_{res}. Our preliminary experiments demonstrate that this strategy enhances robustness during the inference phase.

Overall, our USED model has $(k+1)$ input embeddings, where the first k embeddings with *active* and *blank* states support the predictions of at most k speakers and the $(k+1)$-th embedding E_{k+1} is with the *residual* state. Finally, we shuffle all the embeddings except E_{k+1} to guarantee the model's insensitivity to the order of embeddings. Through the designed embedding processing mechanism, our USED model can predict the outputs from a custom number of speakers. Additionally, the model's robustness is enhanced by employing a random selection strategy for speakers when assigning *blank* state.

F. Separator

The separator is designed to separate speakers based on the embeddings from the embedding assignment module. As shown in Fig. 1, the embedding sequence X firstly undergoes a series of operations, including a layer normalization followed by 1×1 CNN layer, wherein the input channel size is set to $3C$ and the output channel size is C. Then, we introduce N^t Temporal Convolutional Network (TCN) blocks [28] to generate output representations based on the embeddings and speech mixture.

We use TCN as the backbone of our model because they have been widely adopted in speech separation and speaker extraction tasks [9], [28]. The USED model is built on the SpEx+ model, which also utilizes TCN. This choice ensures consistency and enhances performance. In addition, TCN is commonly used for time-domain approaches, avoiding the phase estimation issues found in time-frequency models. Finally, compared to methods like DPRNN [29] and BSRNN [30], Conv-TasNet [28] offers lower computational complexity, reduced memory requirements, and faster training and inference.

Representations are then concatenated and down-sampled via a Conv1D layer with input channel size $(k+1) \times C$ and output channel size C. Finally, the processed representation is forwarded to another N^t TCN blocks to generate representations $\{R_1, ..., R_{N^t}\}$.

Each TCN block comprises a set of N^b TCN layers. The dilated depth-wise separable convolution (D-Conv) shown in Fig. 2b has an exponential growth dilation factor 2^b, where $b \in \{0, ..., N^b - 1\}$. In the initial TCN layer within each block, the input channel size for the 1×1 CNN is $D_{\text{spk}} + C$.

G. Diarization Decoder

The diarization decoder aims to predict the frame-level activity probabilities for each speaker from the N^t TCN blocks' outputs $\{R_1, ..., R_{N^t}\}$ in the separator. Because speaker extraction and diarization require different time resolutions, we employ a Conv1D layer to down-sample the outputs from the TCN block, with input and output channel sizes of C, a kernel size of L^{diar}, and a stride of $L^{\text{diar}}/2$. Subsequently, we use a linear layer followed by a softmax operation to compute the probabilities, denoted as $D^j = \{D_1^j, ..., D_{k+1}^j\}$, which represent the speech activities of the corresponding embeddings. Finally, we get a set of diarization results from each TCN block, denoted as $\{D^1, ..., D^{N^t}\}$. During inference, we only use the diarization result D^{N^t} from the final TCN block as the results to calculate the evaluation metric.

H. Extraction Decoder

The extraction decoder is designed to estimate masks for each speaker and reconstruct the corresponding signals. The mask module of the extraction decoder comprises a Conv1D layer and ReLU activation to generate masks $\{M_1, ..., M_{k+1}\}$, where $M_i \in \mathbb{R}^{T \times C}$. We then obtain the modulated responses i by element-wise multiplication of the mask M_i and the representations X from the speech mixture.

For signal reconstruction, we employ a multi-scale decoder like speech encoder, which consists of three transposed convolutional networks (ConvTran1D), each with a kernel size of L_1^{e}, L_2^{e} and L_3^{e}, and a common stride, $L_1^{\text{e}}/2$. It's used to reconstruct the time-domain signals, $\tilde{s} = \{\tilde{s}_1, ..., \tilde{s}_{k+1}\}$, where $\tilde{s}_i = \{\tilde{s}_i^1, \tilde{s}_i^2, \tilde{s}_i^3\}$ is obtained from three ConvTran1D layers.

As the USED model generates both the extracted speech and speaker diarization results, we propose an interaction module (IM) to use speaker diarization results to refine the generated waveforms. To avoid the gradient from the extraction task having an effect on the diarization task, we first create a clone of $D^{N^t} = \{D_1^{N^t}, ..., D_{(k+1)}^{N^t}\}$ without gradient. Subsequently, we employ interpolation techniques on the diarization output probabilities to ensure length-based uniformity between the two tasks. Following this alignment, a layer comprising a Conv1D layer and ReLU activation is employed to further process the output probabilities and obtain outputs r_i, ensuring that the values at each step are not constrained to be less than 1. The Conv1D layer possesses an input channel size of 1, an output channel size of 1, and a kernel size of L^{im}.

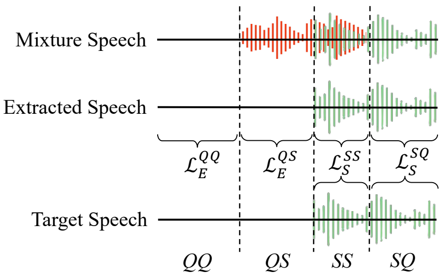


Fig. 3. Illustration of scenario-aware differentiated loss. The speech mixture is segmented according to the four scenarios (*QQ*, *QS*, *SS* and *SQ*).

Finally, the model generates the extracted speeches, $\{\hat{s}_1, ..., \hat{s}_{k+1}\}$, through element-wise multiplication of $\{r_1, ..., r_{k+1}\}$ and the output $\tilde{s}$ from the extraction module separately as

$$\hat{s}_i = \{\hat{s}_i^1, \hat{s}_i^2, \hat{s}_i^3\} \tag{4}$$

where

$$\hat{s}_i^j = \tilde{s}_i^j \otimes r_i \tag{5}$$

It should be noted that, during inference, only $\tilde{s}_1^1, ..., \tilde{s}_k^1$ originating from the first ConvTran1D are utilized for the evaluation of metrics.

I. Loss Function

1) Overall Loss Function: The loss of the USED model can be formulated as follows:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{ext}} + \lambda_2 \mathcal{L}_{\text{diar}} + \lambda_3 \mathcal{L}_{\text{spk}} \tag{6}$$

where λ_1, λ_2 and λ_3 are the weights for speaker extraction loss, speaker diarization loss, and speaker classification loss, respectively.

2) Speaker Extraction: We adopt the SAD loss proposed in [27] to address sparsely overlapped speech as shown in Fig. 3. Scenarios are classified into four distinct classes: *QQ*, *QS*, *SS*, and *SQ*. In the *QQ* scenario, both the target speaker and interference speakers are in a *quiet* state. The *QS* scenario represents a situation where the target speaker is *quiet* while the interference speakers are *speaking*. Conversely, in the *SQ* scenario, the target speaker is *speaking* while the interference speakers are *quiet*. Lastly, the *SS* scenario corresponds to both the target speaker and interference speakers being in a *speaking* state. For *QS* and *QQ*, we use power as the objective for extracted speech $\hat{s}$,

$$\mathcal{L}_{\text{E}} = \sum_{j=1}^{3} 10\mu_j \log_{10}\left(\frac{\|\hat{s}^j\|^2}{T_{\text{se}}} + \epsilon\right) \tag{7}$$

where T_{se} is the duration of $\hat{s}^j$ in seconds, $[\mu_1, \mu_2, \mu_3]$ are the weights for different ConvTran1D layers and ϵ is set to 10^{-6}. For *SS* and *SQ*, we compute the SI-SDR loss between the target speech s and extracted speech $\hat{s}$ as $\mathcal{L}_{\text{S}}$,

$$\mathcal{L}_{\text{S}} = \sum_{j=1}^{3} -10\mu_j \log_{10} \frac{\| \frac{<\hat{s}^j, s> s}{\|s\|^2+\epsilon} \|^2}{\|\hat{s}^j - \frac{<\hat{s}^j, s> s}{\|s\|^2+\epsilon}\|^2 + \epsilon} + \epsilon \tag{8}$$

SAD loss is used to calculate the loss between the extracted speeches $\{\hat{s}_1, .., \hat{s}_{k+1}\}$ generated from the multi-task interaction module and target speeches $\{s_1, ..., s_{k+1}\}$. For clarity, we assume the first m target speeches correspond to the embeddings with *active* state and are the same as $\{c_1, ..., c_m\}$ in Eq. (1). $\{s_{m+1}, ..., s_k\}$ are zero vectors which present silence as the target of embeddings with *blank* state. s_{k+1} is the target speech for residual prediction. In practice, the order is shuffled as introduced in section III-E. Each pair of extracted speech and target speech are firstly segmented according to the four scenarios. Finally, the total speaker extraction loss, i.e. $\mathcal{L}_{\text{ext}}$, is as follows:

$$\mathcal{L}_{\text{ext}} = \frac{1}{k+1} (\sum_{i=1}^{k+1} \alpha \mathcal{L}_{\text{E}_i}^{QQ} + \beta \mathcal{L}_{\text{E}_i}^{QS} + \gamma \mathcal{L}_{\text{S}_i}^{SS} + \delta \mathcal{L}_{\text{S}_i}^{SQ}) \tag{9}$$

where $k+1$ is the total number of outputs and α, β, γ and δ are the weights for different scenarios.

3) Speaker Diarization: To optimize the model for the diarization task, we apply the binary cross-entropy loss (BCE), denoted as $\mathcal{L}_{\text{diar}}^j$ on the output of the N^t TCN blocks,

$$\mathcal{L}_{\text{diar}}^j = \frac{1}{k+1} \sum_{i=1}^{k+1} BCE(y_i^{\text{diar}}, D_i^j) \tag{10}$$

where y_i^{diar} is the ground-truth label for the i-th output. Then, the entire speaker diarization loss can be represented as:

$$\mathcal{L}_{\text{diar}} = \sum_{j=1}^{N^t} \mathcal{L}_{\text{diar}}^j \tag{11}$$

4) Speaker Classification: The speaker encoder is optimized through a cross-entropy loss (CE), denoted as $\mathcal{L}_{\text{spk}}$, for speaker classification during the training process, as

$$\mathcal{L}_{\text{spk}} = \frac{1}{l} \sum_{i=1}^{l} CE(y_i^{\text{spk}}, \hat{y}_i^{\text{spk}}) \tag{12}$$

where y_i^{spk} and $\hat{y}_i^{\text{spk}}$ are the ground truth and predicted speaker ids for speaker i, respectively.

IV. Experimental Setup

A. Datasets

We conduct comprehensive evaluations of our model using diverse datasets, including LibriMix and SparseLibriMix, which are simulated, and CALLHOME, which consists of real recordings.

1) LibriMix: The first dataset, known as LibriMix [21], generates samples derived from LibriSpeech [58] train-clean-100 and test-clean datasets, as well as WHAM! [59], all with a 16kHz sampling rate. LibriMix has been widely used for speech separation [60], [61], speaker extraction [62] and speaker diarization tasks [25], [61]. We merge the Libri2Mix 100h and Libri3Mix 100h datasets, resulting in utterances that may contain either 2 or 3 speakers.

The models are evaluated on both min and max modes. The min mode consists of highly overlapped speech segments, while a larger portion of the max mode consists of non-overlapped speech segments. To ensure compatibility with the speaker extraction task, we have slightly adjusted the data split for training and validation [1], as previously done in related work [62]. Specifically, we follow scripts [2] to prepare speech references for the Libri2Mix dataset and randomly select speech references for each speech mixture in the Libri3Mix dataset.

2) SparseLibriMix: The second dataset employed in our evaluation is the test set from SparseLibriMix [21], which utilizes WHAM! [59] to simulate a noisy acoustic environment. This dataset covers more realistic mixture scenarios, varying from an overlap ratio of 0 to 1.0. It contains 1,000 utterances which have 2 or 3 speakers.

3) CALLHOME: We use the CALLHOME dataset [63] to evaluate the diarization task with real recordings. The CALLHOME dataset is partitioned into two parts based on the Kaldi recipe [3]. The first part, known as Part 1, is utilized for model adaptation, whereas the second part, referred to as Part 2, is used for evaluation. The CALLHOME dataset consists of telephone-channel recordings with 8k sample rate. Besides CALLHOME Part 1, we further add Libri2Mix 100h, Libri3Mix 100h, Libri2Mix 360h and Libri3Mix 360h, which are generated from LibriSpeech [58] train-clean-100 and train-clean-360, as the pre-training dataset. In total, our dataset encompasses approximately 452 hours of data. We then fine-tune the model only with CALLHOME Part 1 and evaluate on CALLHOME Part 2. During inference, we follow the evaluation pipeline of TS-VAD to evaluate our model so that it does not require pre-enrolled speech. Specifically, we extract the speech references based on the diarization result of spectral clustering and then do inference for USED models. The speaker embedding for spectral clustering is extracted from the ECAPA-TDNN model [64] trained on VoxCeleb2 [65] [4].

B. Baseline Configuration

Two of the baseline models, SpEx+ [9], and TS-VAD [14], are implemented by ourselves. During the training phase, we segment the utterances into chunks, each of which spans 4 seconds and is shifted every 2 seconds.

For the SpEx+ baseline, we use the same training configuration as the USED model described in section IV-C. To align the model complexity with that of the USED model, we configure

[1]https://github.com/msinanyildirim/USED-splits
[2]https://github.com/gemengtju/L-SpEx/tree/main/data
[3]https://github.com/kaldi-asr/kaldi/tree/master/egs/callhome_diarization/v2
[4]https://github.com/TaoRuijie/ECAPA-TDNN

the SpEx+ baseline with a total of 8 TCN blocks so that it has a similar number of parameters to the USED model. It is worth noting that the SpEx+ baseline can be regarded as a specialized version of the USED model, as it has only one embedding with *active* state as input without including a diarization module and SAD loss. Our metric for selecting the best checkpoint for the SpEx+ model is SI-SDR.

As for the TS-VAD baseline, we follow [14] to design the model, with several notable modifications. Firstly, we substitute the bidirectional LSTM in the original TS-VAD model with four transformer layers, with sine-cosine positional encoding as input. The model architecture entails two layers dedicated to processing inputs for each speaker individually, followed by a Conv1D layer for down-sampling. Another two layers are used to combine the information from all speakers. For the transformer, we set the embedding dimension to 384 and employ four attention heads. Second, we use a pre-trained speaker encoder, ECAPA-TDNN [64], to extract speaker embeddings. This speaker encoder also takes on the role of replacing the CNN encoder to extract time-level representations from the speech mixture. Finally, we use a similar strategy as stated in section III-E, which randomly selects speaker embeddings from the whole dataset. p_{em} is set to be 0.3. The model is optimized using Adam with batch size 64 for 40k steps. The speech encoder is kept fixed for the first 4,000 steps, and the learning rate follows a polynomial decay schedule with an initial value of 2×10^{-4}, which has a warm-up phase for the first 10% of steps. The DER metric is our criterion for selecting the best checkpoint for the TS-VAD model.

C. USED Model Configuration

The maximum number of speakers k is 3 for evaluation on the LibriMix and SparseLibriMix datasets and in the assessment on the CALLHOME dataset. For the multi-scale speech encoder, C is 256 and the kernel sizes, L_1^e, L_2^e and L_3^e, are 20, 80 and 160. The number of ResNet blocks N^r in the speaker encoder is set to 4, and the dimension of the speaker embedding D_{spk} is 256. Regarding the separator module, we set N^t to 3 and N^b to 8. For the diarization decoder, the CNN layer has a kernel size L^{diar} of 32 and stride $L^{\text{diar}}/2$ of 16. The mask module of the extraction decoder consists of a CNN with a kernel size of 1 and stride 1, and ReLU activation. For the multi-task interaction module, the kernel size L^{im} of the CNN layer is 16. The loss weights for outputs from different ConvTran1D layers, μ_1, μ_2 and μ_3 are 0.8, 0.1 and 0.1 by following [9]. We set 0.001 for α and β, and 1.0 for γ and δ, which are the loss weights in Eq. (9). λ_1, λ_2, and λ_3 are set to 1.0. p_{em} is set to 0.3. p_{sel} and p_{res} are set to 0.5 and 0.9, respectively, when *residual* state is applied. The USED model without *residual* state and with $p_{sel} = 0$ is called as "USED-F(ix)" since it always gets speech references of present speakers during training.

We optimize the model with Adam on 4 GPUs for 100k steps, corresponding to approximately 11 epochs. The utterances are split into chunks with a size of 4 seconds and a shift of 2 seconds. We set the maximum tokens to 260k, which results in processing around 16 seconds of audio for each batch. The learning rate is set to $1e-3$ for all experiments, which is warmed up for the first 10% steps and decayed polynomially for the remaining steps. The overall loss value is our primary metric for selecting the best checkpoint for the USED model.

D. Evaluation Metrics

We use diarization error rate (DER (%)), including speaker confusion (SC (%)), false alarm (FA (%)), and missed detection (MS (%)) to evaluate the speaker diarization performance. Collar tolerance and median filtering are set to 0 seconds and 11 frames for the LibriMix and SparseLibriMix datasets, and 0.25 seconds and 11 frames for the CALLHOME datasets. We report SI-SDR improvement (SI-SDRi (dB)), Power (dB/s), SDR improvement (SDRi (dB)), Short-Time Objective Intelligibility (STOI) [66] and Perceptual Evaluation of Speech Quality (PESQ) [67] for speaker extraction performance. In terms of complexity analysis, we employ Multiply-Accumulate Operations (MACs (G)) to assess the computational cost of the model and perform a calculation of the model's parameter count.

V. Experimental Results and Analysis

A. Results on the LibriMix Dataset

Tables I and II show the results on the LibriMix dataset with max mode and min mode, respectively. As indicated before, min mode mainly contains highly overlapped speech, while a larger proportion of max mode is non-overlapped speech compared to min mode. We provide detailed results under different scenarios for the max mode as shown in Table I. Please note that we use the average duration length that anyone is speaking in seconds as the metric of speaker diarization for the *QQ* scenario.

Our analysis involves comparisons with other popular systems from the literature, including TS-VAD [14], EEND-EDA [13], HuBERT Base [69] and wav2vec 2.0 Base [68] for speaker diarization, SpEx+ [9], and Conv-TasNet [28] for speaker extraction and speech separation, and EEND-SS [25] for joint optimization. For HuBERT Base and wav2vec 2.0 Base, we firstly follow the SUPERB diarization task [61] to train two downstream models for two speakers and three speakers, respectively. Subsequently, we combine the output results of these two models for evaluation.

The proposed USED model demonstrates superior or comparable performance compared to the baseline systems, as indicated by their respective evaluation metrics in both the min and max modes. For speaker diarization, our models exhibit remarkable improvements, with a relative reduction in DER of at least 13% when compared to the prior systems. For speaker extraction, our models achieve a much lower value in terms of power compared to other baseline systems under QQ and QS scenarios in the max mode. This phenomenon implies that speaker diarization and SAD loss effectively mitigate background noise during silent intervals, resulting in enhanced performance.

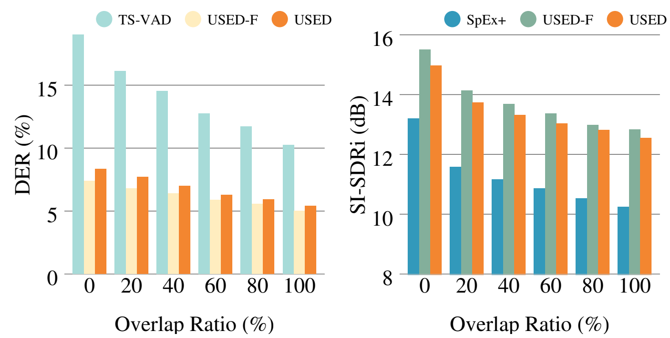
This article has been accepted for publication in IEEE/ACM Transactions on Audio, Speech and Language Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2024.3511268

TABLE I
PERFORMANCE ON THE LIBRIMIX DATASET FOR **MAX MODE**. THE RESULTS OF BASELINE SYSTEMS ARE OBTAINED BY OUR OWN IMPLEMENTATIONS.

Speaker Diarization							
Model	Overall Performance DER↓	MS↓	FA↓	SC↓	SS DER↓	SQ&QS DER↓	QQ Seconds↓
wav2vec 2.0 BASE [68]	7.62	2.28	4.82	0.52	-	-	-
HuBERT BASE [69]	7.56	2.40	4.81	**0.35**	-	-	-
TS-VAD [14]	7.28	3.61	2.78	0.89	5.71	11.63	0.08
USED-F	**4.75**	**2.18**	2.16	0.42	**3.21**	**9.41**	0.05
USED	5.20	2.68	**2.08**	0.43	3.56	10.36	**0.04**
Speaker Extraction							
Model	Overall Performance SI-SDRi↑	SDRi↑	STOI↑	PESQ↑	SS SI-SDRi↑	SQ SI-SDRi↑	QS&QQ Power↓
SpEx+ [9]	9.06	10.11	0.776	1.39	8.32	5.17	55.60
USED-F	**12.70**	**13.22**	**0.844**	**1.46**	**12.00**	9.17	**-24.00**
USED	12.22	12.69	0.836	1.43	11.63	**9.30**	-14.15

TABLE II
PERFORMANCE ON THE LIBRIMIX DATASET FOR **MIN MODE**. † INDICATES THE RESULTS ARE OBTAINED BY OUR OWN IMPLEMENTATIONS.

Speaker Diarization				
Model	DER ↓	MS ↓	FA ↓	SC ↓
EEND-EDA [25]	10.16	-	-	-
TS-VAD † [14]	5.30	2.32	2.83	0.15
EEND-SS [25]	6.27	-	-	-
+ LMF	6.04	-	-	-
USED-F	4.64	1.99	2.55	0.10
USED	**4.57**	**1.94**	**2.54**	**0.09**
Speaker Extraction / Speech Separation				
Model	SI-SDRi ↑	SDRi ↑	STOI ↑	PESQ ↑
Conv-TasNet [25]	7.66	8.71	0.756	-
SpEx+ † [9]	8.76	10.09	0.772	1.44
EEND-SS [25]	9.31	7.50	0.760	-
+ Fusion	9.38	7.59	0.760	-
+ LMF	8.83	9.72	0.767	-
+ LMF + Fusion	8.87	9.77	0.767	-
USED-F	**12.24**	**12.83**	**0.841**	**1.54**
USED	11.98	12.50	0.834	1.44



(a) Speaker Diarization (b) Speaker Extraction

Fig. 4. Performance on the SparseLibriMix for overlap ratio from 0% to 100%. The Left and right bar charts are the results of speaker diarization and speaker extraction tasks, respectively.

B. Results on SparseLibriMix

In addition to the LibriMix dataset, we are also interested in assessing the performance of our model across a range of overlap ratios. Therefore, we conduct an evaluation on SparseLibriMix, as illustrated in Fig. 4. We select TS-VAD and SpEx+ models as the baseline systems. The model trained on the max mode of Libri2Mix and Libri3Mix is evaluated on SparseLibriMix with 2 and 3 speakers. The proposed USED models outperform each task baseline, confirming our models' effectiveness for variable values of overlap ratio.

C. Results on CALLHOME

To compare the performance of diarization on real data, we evaluate our model on the CALLHOME dataset. The results are summarized in Table III, compared with several speaker diarization methods, including two clustering-based methods, AHC clustering [13] and spectral clustering [41], PixIT [44], end-to-end speaker diarization models, EEND-EDA [13], SC-EEND [70] and AED-EEND [71], and the TS-VAD model [14]. Most of those methods use the simulation data from Swichboard-2, Switchboard Cellular, and NIST Speaker Recognition Evaluation [13], which have a variable number of speakers, ranging from one to five. The total number of hours for the pre-training data is around 15,516 hours for those methods. In contrast, we use much less simulation data, which is from LibriMix and only have 2 or 3 speakers. This results in around 452 hours of pre-training data.

Our USED model achieves around 15% relative improvement on overall DER compared to the TS-VAD baseline and performs better with much less pre-training data compared to other methods. We observe that our USED model is more robust in terms of the number of speakers. Results show that although our model only uses simulation data from LibriMix, which has a speech mixture with 2 or 3 speakers, the model still achieves better DER results when the number of speakers is larger than 3. We believe this gain is from the design of the embedding assignment module.

TABLE III
DER (%) RESULTS ON CALLHOME, A DATASET OF REAL RECORDINGS. THE SYMBOL ‡ DENOTES RESULTS OBTAINED FROM OUR OWN IMPLEMENTATIONS, WHILE † INDICATES RESULTS EVALUATED USING THE OPEN-SOURCE MODEL.

Model	Pre-training Data	Number of Speakers 2	3	4	5	6	all
AHC clustering [13]	-	15.45	18.01	22.68	31.40	34.27	19.43
spectral clustering [41]	-	16.05	18.77	22.05	30.46	36.85	19.83
PixIT † [44]	79h	17.02	26.03	27.75	36.38	40.77	24.41
EEND-EDA [13]	15,516h	8.50	13.24	21.46	33.16	40.29	15.29
SC-EEND [70]	15,516h	9.57	14.00	21.14	31.07	37.06	15.75
AED-EEND [71]	15,516h	**6.96**	**12.56**	18.26	34.32	44.52	14.22
TS-VAD ‡ [14]	452h	12.95	14.57	20.26	27.36	32.66	15.51
USED-F	452h	9.09	12.76	**14.67**	**26.96**	26.71	**13.16**
USED	452h	10.23	12.78	14.68	32.02	**25.10**	13.51

TABLE IV
MODEL PERFORMANCE ON THE LBRIMIX DATASET FOR MAX MODE WHEN SETTING DIFFERENT VALUES OF m. RS STANDS FOR RESIDUAL STATE.

m	Model	SI-SDRi ↑	DER ↓
1	SpEx+ [9]	9.06	-
	USED-F	5.71	20.30
	USED	9.73	11.13
	- w/o RS	9.87	11.96
	- $p_{\text{res}} = 1.0$	**10.23**	**10.85**
2	USED-F	9.55	11.27
	USED	11.02	**9.42**
	- w/o RS	10.96	10.98
	- $p_{\text{res}} = 1.0$	**11.08**	10.14
all	TS-VAD [14]	-	7.28
	USED-F	**12.70**	**4.75**
	USED	12.22	5.20
	- w/o RS	11.80	9.46
	- $p_{\text{res}} = 1.0$	11.76	10.38

D. Investigation of USED Models for Generating Results of Different Numbers of Speakers During Inference

In this section, we compare the USED-F model and the USED model with baseline systems when generating results for different numbers of speakers during inference, as shown in Table IV. As introduced in Section III-I2, m is the number of embeddings with an *active* state as the input of the USED model. During inference, m also represents the number of speakers the USED model is required to predict. When m is 1, USED models perform speaker extraction like SpEx+. Under this condition, USED models and SpEx+ have the same input data for a fair comparison. When m is all, it means we provide all speech references for present speakers, and the embedding is V_{em} for *residual* state. Otherwise, we use V_{res} as the embedding for *residual* state.

First, USED models consistently perform much better than the SpEx+ baseline when m is 1, except for the USED-F model. This indicates that even if the USED model only gets one speech reference, it can still outperform the SpEx+ model. When m is 2 or all, the performance of USED in speaker extraction significantly surpasses that of SpEx+. The experimental results show that the USED model demonstrates superior efficiency and performance in speaker extraction tasks, particularly in multi-speaker scenarios. Unlike SpEx+, which requires multiple inferences for each speaker, USED performs inference just once, reducing redundant computations like those in the speech encoder. It can also simultaneously utilize the embeddings of multiple speakers, resulting in more discriminative results, whereas SpEx+ processes one speaker at a time. As the number of speakers increases, USED's performance advantage becomes even more significant.

Second, when extracting results for all speakers, the USED model outperforms TS-VAD, as shown in the experimental results. In addition, the TS-VAD model can only extract results for all speakers simultaneously, while the USED model provides flexibility to extract results for a specific speaker or multiple speakers, which is advantageous in scenarios where obtaining embeddings for all speakers at once is not feasible. Furthermore, the USED model does not require a pre-trained speaker encoder, unlike TS-VAD, which typically relies on one.

Third, we observe that the USED model without *residual* state achieves improvement compared to the USED-F model on both speaker diarization and extraction tasks if m equals 1 or 2. It gets a larger improvement when applying *residual* state with p_{res} of 1.0. However, when m is all, the performance of these two models degrades significantly on the speaker diarization task compared to the USED-F model with the DER increasing from 4.75% to 10.38% and 9.46%, respectively. This degradation can be avoided by setting p_{res} to 0.9 during training. In this way, the USED model can achieve a competitive performance compared to that of the USED-F model if m is all and still significantly outperforms baseline systems when m is 1 or 2.

The performance gap between USED-F and USED models when $m =$ all is due to the fact that the input to the USED-F model consistently includes speaker embeddings for all speakers during training. As a result, USED-F is designed to predict for all speakers simultaneously, making it specifically optimized for scenarios where $m =$ all. However, its performance is less robust in cases where $m \neq$ all (e.g., $m = 1$ or $m = 2$), leading to lower performance compared to the USED model in these situations.

E. Ablation Study: Assessing Single-Task Training for the USED Model

We then compare USED models trained on single task to investigate the impact of each task as illustrated in Table VI. Without loss of generality, we use the USED-F model for

TABLE V
USED-F MODEL PERFORMANCE UNDER DIFFERENT SCENARIOS WHEN USING DIFFERENT LOSS WEIGHTS FOR EACH SCENARIO OF SAD LOSS WITH OR WITHOUT MULTI-TASK INTERACTION MODULE. IM MEANS MULTI-TASK INTERACTION MODULE. THE RESULTS OF THE FIRST LINE ARE FROM THE STANDARD SETTING.

α	β	γ	δ	IM	Speaker Extraction SI-SDRi ↑ *All*	SI-SDRi ↑ *SS*	SI-SDRi ↑ *SQ*	Power ↓ *QS & QQ*	Speaker Diarization DER↓
0.001	0.001	1.0	1.0	✓	12.70	12.00	9.17	-24.00	4.75
0	0	1.0	1.0	✗	12.47	12.02	9.58	55.68	4.80
0.001	0.001	1.0	1.0	✗	12.71	12.00	9.11	-1.93	4.65
0.01	0.01	1.0	1.0	✗	12.50	11.65	8.68	-45.16	4.77
1.0	1.0	1.0	1.0	✗	11.54	10.48	7.42	-79.21	5.57

TABLE VI
ABLATION STUDY OF THE USED MODEL CONFIGURED FOR A SINGLE TASK (ONLY SD OR ONLY SE) ON THE LIBRIMIX DATASET FOR MAX MODE.

Speaker Diarization				
Model	All DER↓	SS DER↓	SQ&QS DER↓	QQ Seconds↓
USED-F	4.75	3.21	9.41	0.05
- Only SD	6.43	4.72	11.76	0.04
Speaker Extraction				
Model	All SI-SDRi↑	SS SI-SDRi↑	SQ SI-SDRi↑	QS&QQ Power↓
USED-F	12.70	12.00	9.17	-24.00
- Only SE	12.46	11.71	9.03	20.84

comparison. We present the results obtained by setting λ_1 or λ_2 to 0, referred to as Only SD (Speaker Diarization) and Only SE (Speaker Extraction), respectively.

The USED-F model achieves a notable reduction of DER with a relative improvement of around 32% in the SS scenario and 20% in the SQ&QS scenarios. The larger improvement in the SS scenario suggests that speaker extraction indeed plays a pivotal role in enhancing the diarization module's ability to handle overlapping speech segments. Furthermore, SI-SDRi increases from 12.46 dB to 12.70 dB for overall performance when compared to the USED-F model with only SE, which may be mainly from the improvement under the QQ scenario where the power reduces from 20.84 dB/s to -24.00 dB/s. This indicates that the speaker diarization task helps surpass the background noise for the speaker extraction task.

F. Investigation of the Relationship between the Multi-Task Interaction Module and the SAD Loss

In this section, we analyze the performance of the USED-F model using different weights for each scenario of SAD loss with or without the multi-task interaction module as illustrated in Table V. For the experiments without the multi-task interaction module, results demonstrate that when increasing the weights α and β for *QQ* and *QS* scenarios, we can get a lower power value, but the SI-SDRi are reduced from 12.02 dB to 10.48 dB for the *SS* scenario and from 9.58 dB to 7.42 dB for the *SQ* scenario. Regarding the performance of speaker diarization, DER results increase when larger values for the weights α and β of *QQ* and *QS* scenarios are used.

TABLE VII
MODEL COMPARISON IN TERMS OF PARAMETERS AND MACS FOR THE TS-VAD MODEL, THE SPEX+ MODEL AND THE USED MODEL. "PARAMS" REFERS TO THE NUMBER OF MODEL PARAMETERS, WHICH IS COUNTED IN MILLIONS (M). †THE MACS OF SPEX+ IS ONLY FOR EXTRACTING ONE SPEAKER'S RESULT.

Model	Params	MACs	Total MACs
TS-VAD [14]	39.50	27.21	158.01
SpEx+ † [9]	16.35	43.60	
USED-F	23.12	96.91	96.91
USED	23.65	119.54	119.54

These observations indicate that the model performance is sensitive to the loss weights under different scenarios. With the help of the multi-task interaction module, we can keep the performance for speaker extraction under *SS* and *SQ* scenarios and speaker diarization and simultaneously achieve a much lower power under the QQ scenario, which decreases from -1.93 dB/s to -24.00 dB/s.

G. Complexity Analysis

In this section, we conduct a complexity analysis of the USED model in comparison to the baseline models, TS-VAD and SpEx+, as shown in Table VII. To ensure a fair comparison, the duration of the speech input for all models is kept consistent, with both the speech mixture and the speech references set to 4 seconds. The experimental results indicate that, compared to individual baseline models, the USED model has more parameters than the SpEx+ model but fewer than the TS-VAD model. However, the computational complexity of the USED model is higher than both SpEx+ and TS-VAD.

It is important to note that the USED model simultaneously addresses two tasks: speaker extraction and diarization, whereas TS-VAD and SpEx+ are designed for a single task, with SpEx+ specifically extracting the speech for only one speaker. Therefore, we calculate the 'Total MACs'. 'Total MACs' refers to the total number of MACs required for the models to process a speech mixture containing three speakers and obtain the corresponding speaker extraction and diarization results for all three speakers. When comparing the 'Total MACs', the 'Total MACs' of the USED and USED-F models are significantly lower than that of the baseline models combined. Specifically, the USED-F model reduces the

TABLE VIII
IMPACT OF HYPERPARAMETER CHOICES FOR EMBEDDING ASSIGNMENT MODULE ON MODEL PERFORMANCE.

m	Metrics	p_{sel}					p_{em}					p_{res}				
		0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
1	SI-SDRi ↑	11.81	10.36	12.12	12.81	12.60	10.77	12.12	12.93	13.09	13.10	13.12	12.99	12.72	13.01	12.12
	DER ↓	6.80	8.79	6.47	5.83	6.19	6.67	6.47	5.87	5.63	5.62	5.75	5.83	6.21	5.76	6.47
2	SI-SDRi ↑	13.28	13.30	13.58	13.62	12.81	13.68	13.58	13.83	13.56	13.42	13.77	13.76	13.83	13.72	13.58
	DER ↓	5.17	5.11	4.98	5.28	9.41	5.01	4.98	5.32	4.97	5.05	5.67	5.15	5.24	4.93	4.98
all	SI-SDRi ↑	14.71	14.45	14.29	14.07	13.10	14.25	14.29	14.31	14.26	14.27	14.31	14.28	14.35	14.25	14.29
	DER ↓	3.87	4.00	4.22	4.41	6.27	4.30	4.22	4.16	4.13	4.18	4.23	4.22	4.16	4.19	4.22

TABLE IX
MODEL PERFORMANCE REGARDING DIFFERENT LOSS WEIGHT VALUES ON THE VALIDATION SET.

λ_1	λ_2	$\lambda_3 = 0.1$		$\lambda_3 = 0.5$		$\lambda_3 = 1.0$	
		SI-SDRi ↑	DER ↓	SI-SDRi ↑	DER ↓	SI-SDRi ↑	DER ↓
0.1	0.1	14.31	4.15	14.27	4.26	14.27	4.15
0.1	0.5	14.01	3.85	13.99	3.76	14.06	3.74
0.1	1.0	13.54	3.85	13.46	4.00	13.32	4.19
0.5	0.1	14.22	5.01	14.13	5.14	14.19	5.01
0.5	0.5	14.16	4.23	14.27	4.26	14.32	4.14
0.5	1.0	14.19	3.97	14.32	3.92	14.28	3.84
1.0	0.1	14.11	5.49	14.15	5.37	14.17	5.47
1.0	0.5	14.15	4.57	14.18	4.55	14.32	4.45
1.0	1.0	14.22	4.21	14.32	4.14	14.29	4.22

total MACs by approximately 39%, while the USED model achieves a reduction of around 24%.

H. Loss Weight Selection

To investigate the impact of different loss weights on the model's performance and explore how to select appropriate values for these weights, we conduct a series of experiments with three loss weights (λ_1, λ_2, and λ_3), as illustrated in Table IX. We evaluate the model's performance on the validation set of the LibriMix dataset with max mode by testing each loss weight with values of 0.1, 0.5, and 1.0. The experimental results indicate that there may be room for improvement in the model's performance regarding the selection of loss weights, although we set the values of all three weights to 1.0 for other experiments. For example, when λ_1, λ_2, and λ_3 are 0.5, 1.0 and 0.5, respectively, the USED model performs slightly better on both the speaker extraction and diarization tasks.

I. Impact on Model Performance of Hyperparameter Tuning for Embedding Assignment Module

In this section, we examine the influence of hyperparameter selection within the embedding assignment module on the overall performance of the model, as shown in Table VIII. Specifically, we focus on three key hyperparameters associated with the embedding assignment module, conducting our analysis using the validation set from the LibriMix dataset in max mode. We assess the impact of varying the hyperparameter values p_{sel}, p_{em}, and p_{res}, which are assigned values of 0.1, 0.3, 0.5, 0.7, and 0.9, while keeping the two other hyperparameters fixed to their default value, across multiple configurations where m is set to 1, 2, and all.

The experimental results demonstrate that the influence of the three hyperparameters on model performance varies depending on the value of m. For instance, for hyperparameter p_{sel}, larger values lead to better results when m equals 1, whereas smaller values yield better performance when m equals all. After carefully evaluating the overall performance across different values of m, we selected a set of parameters with $p_{sel} = 0.5$, $p_{em} = 0.3$, and $p_{res} = 0.9$ for relatively good performance.

VI. CONCLUSION

We propose USED, a unified network for universal speaker extraction and diarization, which integrates speaker extraction and diarization for managing speech mixtures with varying overlap ratios and variable number of speakers. We design an embedding assignment module to support producing results for a variable number of speakers based on speech references and enhance the model's robustness. Additionally, a multi-task interaction module is designed to leverage information from both speaker extraction and diarization tasks. Experimental results demonstrate significant improvements in both highly and sparsely overlapped speech scenarios for the speaker extraction and speaker diarization tasks.

REFERENCES

[1] K. Zmolikova, M. Delcroix, T. Ochiai, K. Kinoshita, J. Černockỳ, and D. Yu, "Neural target speech extraction: An overview," *IEEE Signal Processing Magazine*, vol. 40, no. 3, pp. 8–29, 2023.

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

[3] W. Rao, C. Xu, E. S. Chng, and H. Li, "Target speaker extraction for multi-talker speaker verification," in *Proc. Interspeech 2019*, 2019, pp. 1273–1277.

[4] C. Xu, W. Rao, J. Wu, and H. Li, "Target speaker verification with selective auditory attention for single and multi-talker speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2696–2709, 2021.

[5] K. Žmolíková, M. Delcroix, K. Kinoshita, T. Higuchi, A. Ogawa, and T. Nakatani, "Speaker-aware neural network based beamformer for speaker extraction in speech mixtures," in *Proc. Interspeech 2017*, 2017, pp. 2655–2659.

[6] M. Delcroix, K. Zmolikova, K. Kinoshita, A. Ogawa, and T. Nakatani, "Single channel target speaker extraction and recognition with speaker beam," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5554–5558.

[7] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International conference on machine learning*. PMLR, 2023, pp. 28 492–28 518.

[8] C. Mansfield, S. Ng, G.-A. Levow, R. A. Wright, and M. Ostendorf, "Revisiting parity of human vs. machine conversational speech transcription," in *Interspeech 2021*, 2021, pp. 1997–2001.

[9] M. Ge, C. Xu, L. Wang, E. S. Chng, J. Dang, and H. Li, "SpEx+: A complete time domain speaker extraction network," in *Proc. Interspeech 2020*, 2020, pp. 1406–1410.

[10] C. Xu, W. Rao, E. S. Chng, and H. Li, "SpEx: Multi-scale time domain speaker extraction network," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1370–1384, 2020.

[11] K. Liu, Z. Du, X. Wan, and H. Zhou, "X-SepFormer: End-to-end speaker extraction network with explicit optimization on speaker confusion," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[12] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with permutation-free objectives," in *Proc. Interspeech 2019*, 2019, pp. 4300–4304.

[13] S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, and K. Nagamatsu, "End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors," in *Proc. Interspeech 2020*, 2020, pp. 269–273.

[14] I. Medennikov, M. Korenevsky, T. Prisyach, Y. Khokhlov, M. Korenevskaya, I. Sorokin, T. Timofeeva, A. Mitrofanov, A. Andrusenko, I. Podluzhny, A. Laptev, and A. Romanenko, "Target-speaker voice activity detection: A novel approach for multi-speaker diarization in a dinner party scenario," in *Proc. Interspeech 2020*, 2020, pp. 274–278.

[15] Y. Jiang, Z. Chen, R. Tao, L. Deng, Y. Qian, and H. Li, "Prompt-driven target speech diarization," *arXiv preprint arXiv:2310.14823*, 2023.

[16] Z. Chen, B. Han, S. Wang, and Y. Qian, "Attention-based encoder-decoder end-to-end neural diarization with embedding enhancer," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 1636–1649, 2024.

[17] Z. Chen, T. Yoshioka, L. Lu, T. Zhou, Z. Meng, Y. Luo, J. Wu, X. Xiao, and J. Li, "Continuous speech separation: Dataset and analysis," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7284–7288.

[18] L. E. Shafey, H. Soltau, and I. Shafran, "Joint speech recognition and speaker diarization via sequence transduction," *arXiv preprint arXiv:1907.05337*, 2019.

[19] N. Kanda, Y. Gaur, X. Wang, Z. Meng, Z. Chen, T. Zhou, and T. Yoshioka, "Joint speaker counting, speech recognition, and speaker identification for overlapped speech of any number of speakers," *arXiv preprint arXiv:2006.10930*, 2020.

[20] N. Kanda, Z. Meng, L. Lu, Y. Gaur, X. Wang, Z. Chen, and T. Yoshioka, "Minimum Bayes risk training for end-to-end speaker-attributed ASR," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6503–6507.

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

[22] Özgür Çetin and E. Shriberg, "Analysis of overlaps in meetings by dialog factors, hot spots, speakers, and collection site: insights for automatic speech recognition," in *Proc. Interspeech*, 2006.

[23] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The fifth 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *Proc. Interspeech 2018*, 2018, pp. 1561–1565.

[24] C. Boeddeker, A. S. Subramanian, G. Wichern, R. Haeb-Umbach, and J. L. Roux, "TS-SEP: Joint diarization and separation conditioned on estimated speaker embeddings," *arXiv preprint arXiv:2303.03849*, 2023.

[25] S. Maiti, Y. Ueda, S. Watanabe, C. Zhang, M. Yu, S.-X. Zhang, and Y. Xu, "EEND-SS: Joint end-to-end neural speaker diarization and speech separation for flexible number of speakers," in *2022 IEEE Spoken Language Technology Workshop (SLT)*, 2023, pp. 480–487.

[26] D. Yu, M. Kolbæk, Z.-H. Tan, and J. 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)*. IEEE, 2017, pp. 241–245.

[27] Z. Pan, M. Ge, and H. Li, "USEV: Universal speaker extraction with visual cue," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 3032–3045, 2022.

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

[29] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-Path RNN: Efficient long sequence modeling for time-domain single-channel speech separation," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 46–50.

[30] Y. Luo and J. Yu, "Music source separation with band-split RNN," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1893–1901, 2023.

[31] K. Žmolíková, M. Delcroix, K. Kinoshita, T. Higuchi, A. Ogawa, and T. Nakatani, "Learning speaker representation for neural network based multichannel speaker extraction," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 8–15.

[32] M. Delcroix, K. Zmolikova, T. Ochiai, K. Kinoshita, S. Araki, and T. Nakatani, "Compact network for SpeakerBeam target speaker extraction," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6965–6969.

[33] K. Žmolíková, M. Delcroix, K. Kinoshita, T. Ochiai, T. Nakatani, L. Burget, and J. Černockỳ, "SpeakerBeam: Speaker aware neural network for target speaker extraction in speech mixtures," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 800–814, 2019.

[34] Q. Wang, H. Muckenhirn, K. Wilson, P. Sridhar, Z. Wu, J. R. Hershey, R. A. Saurous, R. J. Weiss, Y. Jia, and I. L. Moreno, "VoiceFilter: Targeted voice separation by speaker-conditioned spectrogram masking," in *Proc. Interspeech 2019*, 2019, pp. 2728–2732.

[35] M. Borsdorf, C. Xu, H. Li, and T. Schultz, "Universal speaker extraction in the presence and absence of target speakers for speech of one and two talkers," in *Proc. Interspeech 2021*, 2021, pp. 1469–1473.

[36] Z. Zhang, B. He, and Z. Zhang, "X-TaSNet: Robust and accurate time-domain speaker extraction network," in *Proc. Interspeech 2020*, 2020, pp. 1421–1425.

[37] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR – half-baked or well done?" in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 626–630.

[38] S. H. Shum, N. Dehak, R. Dehak, and J. R. Glass, "Unsupervised methods for speaker diarization: An integrated and iterative approach," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2015–2028, 2013.

[39] G. Sell and D. Garcia-Romero, "Speaker diarization with PLDA i-vector scoring and unsupervised calibration," in *2014 IEEE Spoken Language Technology Workshop (SLT)*, 2014, pp. 413–417.

[40] M. Senoussaoui, P. Kenny, T. Stafylakis, and P. Dumouchel, "A study of the cosine distance-based mean shift for telephone speech diarization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 1, pp. 217–227, 2014.

[41] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, "Speaker diarization with LSTM," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5239–5243.

[42] S. Wang, Z. Chen, K. A. Lee, Y. Qian, and H. Li, "Overview of speaker modeling and its applications: From the lens of deep speaker representation learning," *arXiv preprint arXiv:2407.15188*, 2024.

[43] Y.-X. Wang, J. Du, M. He, S.-T. Niu, L. Sun, and C.-H. Lee, "Scenario-dependent speaker diarization for DIHARD-III challenge," in *Proc. Interspeech 2021*, 2021, pp. 3106–3110.

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

[45] Y. Bando, T. Nakamura, and S. Watanabe, "Neural blind source separation and diarization for distant speech recognition," *arXiv preprint arXiv:2406.08396*, 2024.

[46] S. Wisdom, E. Tzinis, H. Erdogan, R. Weiss, K. Wilson, and J. Hershey, "Unsupervised sound separation using mixture invariant training," *Advances in neural information processing systems*, vol. 33, pp. 3846–3857, 2020.

[47] K. Kinoshita, M. Delcroix, and N. Tawara, "Advances in integration of end-to-end neural and clustering-based diarization for real conversational speech," *arXiv preprint arXiv:2105.09040*, 2021.

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

[49] Y. Bando, K. Sekiguchi, Y. Masuyama, A. A. Nugraha, M. Fontaine, and K. Yoshii, "Neural full-rank spatial covariance analysis for blind source

separation," *IEEE Signal Processing Letters*, vol. 28, pp. 1670–1674, 2021.

[50] Y. Bando, T. Aizawa, K. Itoyama, and K. Nakadai, "Weakly-supervised neural full-rank spatial covariance analysis for a front-end system of distant speech recognition," in *Interspeech 2022*, 2022, pp. 3824–3828.

[51] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[52] He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[53] J. S. Chung, J. Huh, S. Mun, M. Lee, H. S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In defence of metric learning for speaker recognition," *arXiv preprint arXiv:2003.11982*, 2020.

[54] T. Zhou, Y. Zhao, and J. Wu, "ResNeXt and Res2Net structures for speaker verification," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 301–307.

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

[56] H. Zeinali, S. Wang, A. Silnova, P. Matějka, and O. Plchot, "BUT system description to VoxCeleb speaker recognition challenge 2019," *arXiv preprint arXiv:1910.12592*, 2019.

[57] S. Wang, K. Zhang, S. Lin, J. Li, X. Wang, M. Ge, J. Yu, Y. Qian, and H. Li, "WeSep: A scalable and flexible toolkit towards generalizable target speaker extraction," in *Interspeech 2024*, 2024, pp. 4273–4277.

[58] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: An ASR corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[59] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. L. Roux, "WHAM!: Extending Speech Separation to Noisy Environments," in *Proc. Interspeech 2019*, 2019, pp. 1368–1372.

[60] K. Li, R. Yang, and X. Hu, "An efficient encoder-decoder architecture with top-down attention for speech separation," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=fzberKYWKsI

[61] S. wen Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K. tik Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H. yi Lee, "SUPERB: Speech processing universal performance benchmark," in *Proc. Interspeech 2021*, 2021, pp. 1194–1198.

[62] M. Ge, C. Xu, L. Wang, E. S. Chng, J. Dang, and H. Li, "L-SpEx: Localized target speaker extraction," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7287–7291.

[63] P. Mark and A. Martin, "2000 NIST speaker recognition evaluation," 2001. [Online]. Available: https://catalog.ldc.upenn.edu/LDC2001S97

[64] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Interspeech 2020*, 2020, pp. 3830–3834.

[65] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Interspeech 2018*, 2018, pp. 1086–1090.

[66] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 4214–4217.

[67] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, vol. 2, 2001, pp. 749–752 vol.2.

[68] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 12 449–12 460. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/92d1e1eb1cd6f9fba3227870bb6d7f07-Paper.pdf

[69] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[70] Y. Fujita, S. Watanabe, S. Horiguchi, Y. Xue, J. Shi, and K. Nagamatsu, "Neural speaker diarization with speaker-wise chain rule," *arXiv preprint arXiv:2006.01796*, 2020.

[71] Z. Chen, B. Han, S. Wang, and Y. Qian, "Attention-based encoder-decoder network for end-to-end neural speaker diarization with target speaker attractor," in *Proc. INTERSPEECH 2023*, 2023, pp. 3552–3556.