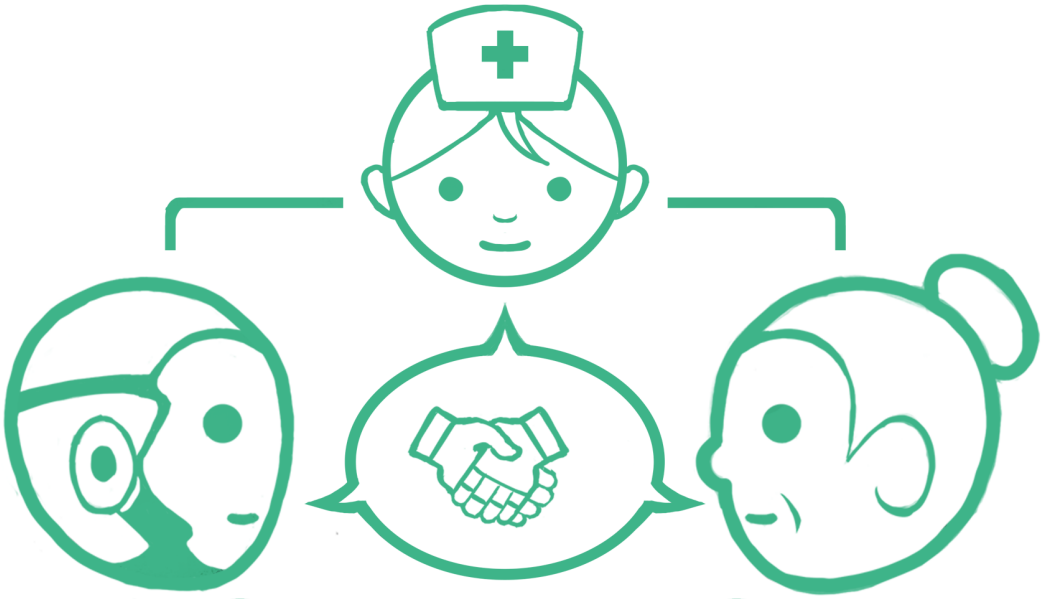


Deliverable D3.4: Audio speaker diarisation and extraction with a moving robot

Due Date: 07/04/2023

Main Author: Sharon Gannot (BIU)

Contributors: -

Dissemination: Public Deliverable

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

DOCUMENT FACTSHEET

Deliverable	D3.4: Audio speaker diarisation and extraction with a moving robot
Responsible Partner	BIU
Work Package	WP3: Robust Audio-visual Perception of Humans
Task	T3.3: Extraction of Desired Sources (Moving Robot)
Version & Date	07/04/2023
Dissemination	Public Deliverable

CONTRIBUTORS AND HISTORY

Version	**Editor**	**Date**	**Change Log**
1	BIU	07/04/2023	First Draft

APPROVALS

Authors/editors	Sharon Gannot (BIU)
Task Leader	BIU
WP Leader	BIU

This project has received funding from the European Union's Horizon 2020
Research and Innovation Programme under Grant Agreement No. 871245.

Contents

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

Executive Summary

Deliverable 3.4 reports the progress on task T3.3 on Audio speaker diarisation and extraction with a moving robot, which is part of WP3: Robust Audio-visual Perception of Humans. The goal of task 3.3 is to provide several separated audio streams to be transcribed by the automatic speech recognition (ASR) and fed to the multi-party conversational system that will be deployed on ARI, the robotic platform designed by PAL Robotics for the SPRING project.

The main achievements reported in this document are:

1. Single-microphone speaker extraction algorithm, using a reference utterance of the desired speaker, which is an improved version of the speaker extraction method reported in D3.3.
2. A single-microphone speaker separation algorithm, based on a temporal convolutional network (TCN) module.

Both methods are extensively tested with common databases and also with real recordings from ARI. word error rate (WER) improvements are also reported.

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

1 Introduction

This deliverable is part of WP3 of the H2020 SPRING project. The objective of WP3 is "the robust extraction, from the raw auditory and visual data, of users' low-level characteristics, namely: position, speaking status and speech signal." Following this objective, WP3 has two main outcomes:

1. The Multi-Person Tracking module, jointly exploiting auditory and visual raw data to detect, localise and track multiple speakers (corresponds to T3.1).
2. The Diarisation and Separation and the Speech Recognition modules, extracting the desired speaker(s) from a speech dynamic mixture and recognising the speech utterances from the separated sources, for a static T3.2 and a moving T3.3 robot.

In this deliverable, we report on two algorithms for extracting the speaker of interest from a mixture of multiple speakers.

Single microphone speaker extraction: This algorithm uses a reference signal, which is an utterance from one of the speakers in the mixed signal. Such a reference can be obtained from segments for which only a single speaker is active. This is an extended version of the algorithm reported in D3.3, with two main changes: 1) an iterative separation module, and 2) an additional residual interference suppression and dereverberation module. [1]

Single-microphone speaker separation: This algorithm applies a TCN module for separating the sources. The encoder and decoder are implemented as short-time Fourier transform (STFT) and inverse short-time Fourier transform (iSTFT). Simultaneously, the algorithm infers the activity patterns of the speakers.[2]

From a system perspective, the separation/extraction algorithms provide multiple outputs of separated (and denoised) signals. An ASR is applied to each signal and a text stream is published for further processing by the dialogue system.

[1] Will be available on July 1, 2023, at https://gitlab.inria.fr/spring/wp3_av_perception/1ch_speaker_extraction
[2] Available at https://gitlab.inria.fr/spring/wp3_av_perception/audio_separation

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

2 Literature Survey

Speaker separation or speaker extraction of a desired speaker from a mixture of overlapping speakers using only a single microphone is a cumbersome task, particularly in noisy and reverberant environments.

There has been significant progress in the single-microphone blind source separation (BSS) domain in the past years. The Conv-Tasnet [15] and the dual-path recurrent neural network (DPRNN) [14], are both applied in the time domain with similar encoder-masking-decoder architecture.

Other works that followed this approach were presented [2, 3, 16, 18, 20, 21, 23, 30], demonstrating a considerable improvement in the separation results. The SepFormer was introduced in [20] leveraging the benefits of attention layers, which led to a significant improvement in performance and to state-of-the-art (SOTA) results. In [30] a two-phase DNN is jointly trained, where the first phase learns the embedding of the speakers and the second actually applies the separation operation. An efficient convolutional neural network (CNN)-based model, denoted successive downsampling and resampling of multi-resolution features (SuDoRmRf), was presented in [21], demonstrating high separation capabilities.

Most of the above-mentioned BSS models were trained and tested on clean and anechoic mixtures. Such acoustic conditions can hardly be met in reality. Several algorithms [2, 20, 21, 30] were also trained on reverberant data without any changes in their architecture. Cord-Landwehr et al. showed in [4] that despite the significant improvement achieved in clean conditions, only marginal improvements can be obtained in realistic reverberant and noisy conditions.

Another major obstacle in applying BSS algorithms, specifically, in the context of deep neural network (DNN)-based training, is the permutation ambiguity problem. The utterance-level permutation invariant training (uPIT) was introduced in [12] and paved the way to alleviate this problem.

In [24], a two-stage model was proposed in which a variant of the weighted prediction error (WPE) method was applied to the output of the first stage to produce the final separated signals. In [7], a deep computational auditory scene analysis (CASA) model was proposed for separating reverberant signals. The reverberant mixture was first separated frame by frame using an estimated mask, and then the frames were reorganized using an embedding vector and the K-means algorithm to fix the permutation of the frames. A subsequent module was finally applied to the reverberant signals to produce the final clean target speakers.

Given a reference signal of the desired speaker turns the BSS problem into an extraction problem, in which the permutation problem is inherently alleviated. The SpeakerBeam algorithm, introduced in [32], estimates a mask for the desired speaker in the spectral domain using the spectrum of the reference signal. While magnitude-domain processing might be sufficient in clean and anechoic conditions, it might be insufficient in noisy and reverberant conditions. In [6], this model was improved by using the time-domain signal, as it allows the exploitation of the entire signal information. A similar approach was presented in [27], where the i-vector [5] of the reference signal was used as the embedding of the desired speaker. In [28], a multi-task training procedure was proposed in which a speaker classification task is carried out in parallel for improving the embedding of the desired speaker.

This project has received funding from the European Union's Horizon 2020
Research and Innovation Programme under Grant Agreement No. 871245.

3 A Two-Stage Speaker Extraction Algorithm under Adverse Acoustic Conditions using a Single-Microphone

In this work, we present a two-stage method for speaker extraction under reverberant and noisy conditions. Given a reference signal of the desired speaker, the clean, but still reverberant, desired speaker is first extracted from the noisy-mixed signal. In the second stage, the extracted signal is further enhanced by joint dereverberation and residual noise and interference reduction. The proposed architecture comprises two sub-networks, one for the extraction task and the second for the dereverberation task. We present a training strategy for this architecture and show that the performance of the proposed method is on par with other SOTA methods when applied to the WHAMR! dataset. Furthermore, we present a new dataset with more realistic adverse acoustic conditions and show that our method outperforms the competing methods when applied to this dataset as well.

3.1 Preface

We address the challenge of extracting a single participant from a mixture of two speakers acquired by a single microphone, given a prerecorded utterance of the speaker to be extracted.

Time domain processing, despite its clear advantages as surveyed above, ignores the time-frequency patterns typical to speech signals. In our prior work [8], already reported in D3.3, a fully convolutional Siamse-Unet architecture was proposed. The algorithm is applied in the STFT domain to the real-imaginary (RI) representation of the signals, while the loss is applied in the time domain, exploiting the entire signal, on the one hand, and leveraging its spectral patterns, on the other hand. Yet, the performance of this approach is insufficient in adverse acoustic conditions.

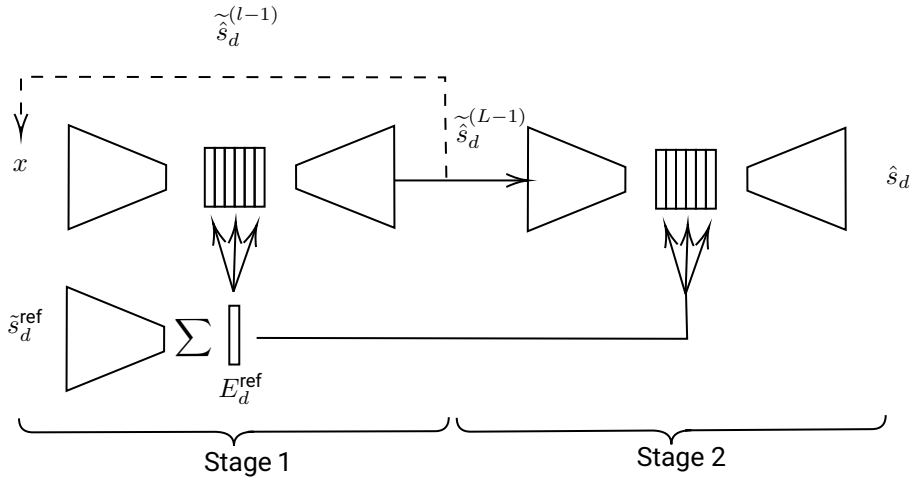


Figure 3.1: Block diagram of the proposed two-stage architecture. In the first iteration, the model takes the given observations as input. For subsequent iterations, the output of the previous iteration is used as input instead of the mixture. The network is using skip connections between the mixture encoder and decoder (not shown explicitly in the diagram). No skip connections from the reference encoder are implemented. The two encoders share the same weights. The arrows denote the element-wise multiplication of the reference embedding with the embedding of each frame. Only the output of the final iteration is used as input to the second stage.

In the current contribution, we extend the work in [8] and present a two-stage algorithm to extract a desired speaker from a mixture of two signals under reverberant and noisy conditions. We split the extraction task into two stages. In the first stage, given the noisy and reverberant mixture and the reference signals, a Siamse-Unet architecture is applied to extract the *reverberant* desired speaker. The encoders used for both the mixture and the reference signals

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

are identical, thus the resulting outputs have matching dimensions. While the mixture encoder preserves the frame dimensions, which is essential for the mixture processing, the reference encoder aims to exclusively represent the desired speaker's identity while ignoring the content of the utterance. To achieve this outcome, we average the reference embedding over the frame dimension. The reference embedding vector is finally multiplied with each of the frames in the mixture embedding. The outcome of this multiplication is used as an input to the decoder, which in turn extracts the reverberant desired speaker.

We show that training this stage in an iterative manner is beneficial.

In the second stage, an additional Unet model is applied to dereverberate and further enhance the output of the first stage. Similarly, the encoder output preserves the frame size of its input signal. The resulting embedding is multiplied by the embedding of the reference from the first stage. The second decoder is finally applied to produce the desired *clean and dereverberated* signal.

Furthermore, in this paper, we introduce a new simulated dataset with more realistic conditions than the WHAMR! dataset, and show that our model outperforms other SOTA models on both the WHAMR! dataset and the new, more challenging, dataset.

A block diagram of the new system is depicted in Fig. 3.1.

3.2 Problem Formulation

The signal $x(t)$, captured by a single microphone, is a combination of Q concurrent speakers, represented by:

$$x(t) = \sum_{q=1}^{Q} \{s_q * h_q\}(t) + v(t) \quad t = 0, 1, \ldots, T-1 \tag{3.1}$$

where $s_q(t)$ is the signal of the qth speaker, $h_q(t)$ is the room impulse response (RIR) between the qth speaker position and the microphone position, and $v(t)$ is an additive noise. In a noise-free, non-reverberant environment, $h_q(t)$ is dominated by the first arrival, and $v(t) = 0$ for all q.

In the STFT domain, the microphone signal can be approximately expressed as:

$$x(n,k) = \sum_{q=1}^{Q} s_q(n,k) h_q(n,k) + v(n,k) \tag{3.2}$$

where $n = 0, 1, \ldots, N-1$ and $k = 0, 1, \ldots, K-1$ represent the time-frame and frequency-bin indexes, respectively, and N and K are the total number of time-frames and frequency bands, respectively.

This paper focuses on the case where there are only two concurrent speakers, namely $Q = 2$, referred to as the desired speaker $s_d(n,k)$ and the interference speaker $s_i(n,k)$. The reverberant desired signal is defined as $\tilde{s}_d(n,k) = s_d(n,k)h_d(n,k)$. The reference signal is denoted $s_d^{\text{ref}}(n,k)$. We aim at the extraction of the desired speaker signal, $\hat{s}_d(n,k)$, using the mixed signal $x(n,k)$, and a reverberant reference signal, $\tilde{s}_d^{\text{ref}}(n,k) = s_d^{\text{ref}}(n,k)h_d^{\text{ref}}(n,k)$.

3.3 Proposed Model

3.3.1 Architecture and Training Procedure

Our model is composed of two sub-stages. The first is a Simase-Unet, which consists of three parts: two encoders and a decoder. We share weights between the encoders to encourage joint embeddings of both the mixture and the reference signals in the same latent space. The encoder architecture consists of several convolution layers followed by two-dimensional batch normalization and a 'Relu' function (similar to the one introduced in [8]). Next, we combine the dimensions of the channels and frequencies and employ a fully-connected layer to reduce the dimensions. After this step, we apply a single transformer-encoder layer. The decoder architecture consists of six transformer-encoder layers, followed by fully-connected (FC) layer to restore the original dimension. Then transpose-convolution layers are employed to adapt to the convolution layers in the encoder, enabling the application of skip connections as required. A transformer-encoder layer is subsequently applied after all the steps mentioned above. We repeat the first stage several times to further enhance the extraction process. In the first iteration, the mixture signal is processed, while in the subsequent iterations, the separated (but still reverberant) signals from the previous iteration are processed. Formally, the process can be expressed as:

$$\text{Input}^{(\ell)} = \begin{cases} x(n,k) & \ell = 0 \\ \hat{s}_d^{(\ell-1)}(n,k) & \ell > 0 \end{cases}$$

where $\ell = 0, ..., L-1$ is the iteration index. By repeating this process for L iterations, we obtain L estimates of $\tilde{s}_d(n,k)$, which are all used to train the entire model.

The second stage of the model uses the same architecture as the first stage. Our empirical results showed that using the reveberant reference signal in the second phase can improve the results. Rather than passing the reference signal again through an encoder, we can simply use the learned embedding vector from the first stage.

Alternative ways for integrating the information from the reference signal are described in [32], including concatenation, addition, and multiplication, the latter achieving the best results. To obtain a single vector that represents the speaker's identity, we average across the frame dimensions of the reference embedding, thus ignoring the temporal information and emphasizing the speaker's identity. The final embedding vector is denoted E_d^{ref}. Unlike [8], in the Unet architecture, skip connections are only implemented from the mixture encoder and not from the reference encoder. Instead, we only use the output of the last layer of the reference encoder in the bottleneck stage. While most single microphone DNN-based algorithms apply a masking operation to the mixture signal, the proposed scheme is trained to directly estimate the time frequency (TF) representation of the target source.

The two sub-stages are trained together in an end-to-end manner, while the first stage feeds the second phase with an estimate of the last iteration of the first stage and the reference embedding. A block diagram of the entire model is shown in Fig. 3.1.

3.3.2 Features

In this work, we adopted the RI components of the STFT as both the input features of the model and its output. The model is trained with the scale-invariant signal-to-distortion ratio (SI-SDR) loss function, which is sensitive to phase distortion. Using the RI features may alleviate such problems (see discussion in [8]).

3.3.3 Objectives

As mentioned above, we use the SI-SDR loss function to train our model. The loss is formulated as

$$\text{SI-SDR}\left(s,\hat{s}\right) = 10\log_{10}\left(\frac{\left\|\frac{\langle \hat{s},s\rangle}{\langle s,s\rangle}s\right\|^2}{\left\|\frac{\langle \hat{s},s\rangle}{\langle s,s\rangle}s-\hat{s}\right\|^2}\right). \tag{3.3}$$

The model is trained using all output signals, namely, $\hat{s}_d$ and $\hat{\tilde{s}}_d^{(\ell)}, \ell = 0, \ldots, L-1$:

$$\mathcal{L}_{\text{SISDR}_d} = \sum_{\ell=0}^{L-1} \text{SI-SDR}\left(\tilde{s}_d, \hat{\tilde{s}}_d^{(\ell)}\right) + \text{SI-SDR}\left(s_d, \hat{s}_d\right). \tag{3.4}$$

For the extraction task to be successful, the network must be able to learn a unique embedding for each speaker to prevent errors in identifying the correct speaker. To achieve this goal, an additional, triplet loss function, was implemented:

$$\text{TRIPLET}(a,p,n) = \max(\text{cd}(a,p) - \text{cd}(a,n) + m, 0) \tag{3.5}$$

where a is the anchor input, p is the positive input and n is the negative input, with p closer to a than n. The function $\text{cd}(\cdot)$ is the cosine distance and m is a margin hyperparameter. The triplet loss function encourages the distance between the anchor and the positive inputs to be smaller than the distance between the anchor and negative inputs, by a margin of at least m. In our case, we would like the embedding of the reverberant reference to be as close as possible to the embedding of the output of the first phase (namely, the estimated desired and reverberant speaker), and as far as possible from the embedding of the reference of the second speaker. In explicit terms:

$$\mathcal{L}_{\text{TRIPLET}_d} = \text{TRIPLET}(E_{\hat{\tilde{s}}_d}, E_d^{\text{ref}}, \overline{E_d^{\text{ref}}}) \tag{3.6}$$

where $E_{\hat{\tilde{s}}_d}$ is obtained by passing $\hat{\tilde{s}}_{d_{L-1}}$ through the encoder of stage 1 and $\overline{E_d^{\text{ref}}}$ is the embedding of the reference of the interference signal.

During training, we encountered a convergence problem when using both loss functions simultaneously. To address this issue, we implemented a warm-up training procedure in which the network is initially trained using only the SI-SDR loss, and the triplet loss is added at a later stage in the training process. This approach successfully resolved the convergence issues.

In an effort to improve the training process, we alternated the desired and interference signals within each training batch, while maintaining consistency in the mixture employed. That is, inserting the mixture signal with the reference

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

Table 3.1: Noisy reverberant data specification.

Room dim. [m]	H_x	$U[4, 8]$
	H_y	$U[4, 8]$
	H_z	$U[2.5, 3]$
Reverb. time [sec]	T_{60}	$U[0.2, 0.6]$
Mic. Pos. [m]	x	$\frac{H_x}{2} + U[-0.5, 0.5]$
	y	$\frac{H_y}{2} + U[-0.5, 0.5]$
	z	1.5
Sources Pos. [°]	θ	U[0,180]
Sources Distance [m]		$1 + U[-0.5, 0.5]$

signal of one of the speakers and then repeating the process with the reference of the other speaker in the same batch, and summing the losses for both speakers. In short, the overall loss function takes the following form:

$$\mathcal{L} = (\mathcal{L}_{\text{SISDR}_d} + \mathcal{L}_{\text{SISDR}_i})/2 \ + \alpha \cdot \mathbb{1}_{\text{warm-up}} \cdot (\mathcal{L}_{\text{TRIPLET}_d} + \mathcal{L}_{\text{TRIPLET}_i})/2 \tag{3.7}$$

where α represents a hyperparameter, and the indicator function $\mathbb{1}_{\text{warm-up}}$ determines the point at which the triplet objective function should be taken into consideration in the training process.

3.4 Experimental Study

3.4.1 Datasets

We used the WHAMR! dataset to train our model. This dataset is created by taking the WSJ0-2Mix dataset [10] and modifying it by incorporating environmental noise from the WHAM dataset [26] and reverberation. To adapt the dataset to the extraction task, we modified it in the following manner. For each speaker included in the mixture, we selected a different utterance and convolve it with the same room impulse response (RIR) used to generate the mixed signal, namely $h_d^{\text{ref}} = h_d$. This procedure reflects the fact that in a typical conversation, segments in which only a single speaker is active can always be found. However, it is implicitly assumed that the scenario is static, hence that the RIR does not significantly change during the entire conversation.

We note that, according to our tests, the reverberation level in the WHAMR! dataset does not exceed 600 milliseconds, in contradiction to the reported reverberation level, which is in the range of $[0.2, 1]$.[1]

The dataset includes 20,000 signals for training, 5,000 for validation, and 300 for the test phase, and it uses the 'min' and '8k' sampling rate configuration. (With 'min' setting the longer target is truncated to match the length of the shorter target.)

In addition to WHAMR!, we generated a new dataset for the purpose of enriching the data. This is equivalent to *dynamic mixing* training, which randomly generates the mixture from the existing speakers during training. We also took speakers from the WSJ0 corpus, along with noise from the WHAM and the reverberation generated from an RIR generator [9] with parameters listed in Table 3.1.

During training, each signal is truncated to a variable length between 2 to 5 seconds. Since we are using a Siamese architecture, the mixture and the reference signal must have the same length. If the reference signal is longer, it will be truncated, and if it is shorter, it will be duplicated until it is the same length as the mixture.

3.4.2 Algorithm Settings

The frame size of the STFT is 256 samples with 50% overlap. Due to the symmetry of the discrete Fourier transform (DFT) only the first half of the frequency bins are used. The value of α was empirically set to 2, emphasizing the triplet loss due to the significant difference in scales between the two objective functions. The triplet loss margin was set to $m = 0.5$.

The number of iterations for the first phase was chosen as $L = 2$, because there was minimal improvement when increasing the number from 2 to 3 iterations, while a noticeable improvement was observed between 2 iterations to no iterations, $L = 1$.

[1] Due to space constraints, we will not give a detailed analysis of the dataset in the current contribution.

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

In the training procedure, we used the Adam optimizer [11]. The learning rate was set to 0.001 and the training batch size to 6. The weights are randomly initialized, and the lengths of the signals were randomly changed at each batch.

3.4.3 Evaluation Measures

To evaluate the proposed algorithm we use five evaluation measures: SI-SDR, signal to interference (SIR), signal direction recognition (SDR), short-term objective intelligibility (STOI), and perceptual evaluation of speech quality (PESQ). While the first three are used as a measurement of the quality of the speaker separation, the last two give an indication of the audio intelligibility and quality.

The proposed algorithm is compared to the current SOTA separation methods, i.e., the Sepformer [20] and the SuDoRmRf [21]. These are time-domain blind source separation masking-based methods. We decided to compare our method with separation methods rather than extraction methods since these are the most effective methods in the field.

3.4.4 Results

The results for the WHAMR! dataset are depicted in Table 3.2. Our model achieves an SI-SDR of 9.67 dB, SDR of 10.88 dB, and SIR of 24.2 dB. It is evident that our proposed method outperforms the SOTA methods in almost all measures. In addition, the method also achieves the best scores for the intelligibility measure (STOI) and the quality measure (PESQ), with scores 92% and 2.72, respectively.

The new dataset imposes a greater challenge on the extraction algorithm, as evidenced by the lower scores in Table 3.3 for all measures, compared to the scores obtained on the WHAMR! dataset, as reported in Table 3.2. While the absolute separation results obtained for the new dataset are lower, the improvement in terms of SI-SDR is 14.2 dB, which is very high and significantly outperforms the competing methods. The intelligibility results (90.2%) are on par with the results obtained for the WHAMR! dataset.

Table 3.2: Results for WHAMR! dataset

Model	SI-SDR	SDR	SIR	STOI	PESQ
Unprocessed	-3.84	-0.59	0.19	65.3	1.51
SuDoRmRf [21]	8.13	10.7	23.7	90.2	2.5
Sepformer [20]	8.86	10	**25**	91.3	2.57
Proposed	**9.67**	**10.88**	24.2	**92**	**2.72**

Table 3.3: Results for the new dataset

Model	SI-SDR	SDR	SIR	STOI	PESQ
Unprocessed	-7.99	-0.79	0.12	52.5	1.54
SuDoRmRf [21]	1.7	3.46	15.8	69.9	2.1
Sepformer [20]	1.89	4.82	18.48	68.8	2.05
Proposed	**6.21**	**7.98**	**22.14**	**90.2**	**2.62**

3.4.5 Ablation Study

We present an ablation study for our model. We examined four different configurations:

1. One iteration in the first stage. The loss function for the desired source is given by:

$$\mathcal{L}_{\text{SISDR}_d} = \text{SI-SDR}\left(\tilde{s}_d, \hat{\tilde{s}}_d^{(L-1)}\right) + \text{SI-SDR}\left(s_d, \hat{s}_d\right) \tag{3.8}$$

with $L = 1$, and the overall loss is given by $\mathcal{L} = (\mathcal{L}_{\text{SISDR}_d} + \mathcal{L}_{\text{SISDR}_i})/2$. Triplet loss is not applied.

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

2. Two iterations in the first stage. The SI-SDR loss is only applied to the final output $\hat{s}_d^{(L-1)}$, i.e. $L = 2$ in (3.8). Triplet loss is not applied.
3. The SI-SDR loss is applied to all intermediate results $\ell = 0, \ldots, L-1$, as in (3.4), with $L = 2$. Triplet loss is not applied.
4. The full implementation of the proposed model with all its components active.

Table 3.4 depicts the breakdown of the results for the WHAMR! and the new datasets. It is evident that each additional component enhances the quality of the network output for both datasets. In total, the SI-SDR measure improved from 8.62 dB to 9.67 dB for the WHAMR! dataset and from 5.45 dB to 6.21 dB for the new dataset. Respectively, STOI improved from 90.4% to 92% for WHAMR!, and from 88% to 90.2% for the new dataset

Training the model to accurately identify the intended speaker from a mixture is challenging in speaker extraction, particularly in reverberant conditions and when the speakers have similar voices. This may result in the extraction of the incorrect speaker or a permutation between the output signals. To address this issue, the triplet loss was added. Our experiments showed that the addition of the triplet loss alleviated such permutation problems.

Table 3.4: Ablation Study for all 4 configurations.

	WHAMR!		New	
Conpng.	SI-SDR	STOI	SI-SDR	STOI
1)	8.62	90.4	5.45	88
2)	9.13	91	5.71	88.9
3)	9.26	91.8	6.02	**90.2**
4)	**9.67**	**92**	**6.21**	**90.2**

3.5 Conclusions

We have proposed a two-stage approach for speaker extraction under reverberant conditions. The first stage separates the desired and yet reverberated speaker, while the second stage reduces reverberation and further enhances separation quality. Our results indicate that our model performs comparably or better than current state-of-the-art separation methods, with the added benefits of faster and more consistent training. Furthermore, an ablation study identifies the role of the various components in improving performance.

This project has received funding from the European Union's Horizon 2020
Research and Innovation Programme under Grant Agreement No. 871245.

4 Single-Microphone Speaker Separation and Voice Activity Detection in Noisy and Reverberant Environments

The recent Conv-Tasnet method [15] was shown to be very effective in separating speech sources in an anechoic environment. The method is implemented in the time domain and directly applied to the raw audio signal, using a learnable encoder-decoder. In this architecture the chosen length of the time frames is relatively short, hence limiting the performance in high reverberation.

In the current work, we present a new, fully-supervised, speech separation algorithm for noisy and reverberant environments that uses a single microphone. This network adopts the Conv-Tasnet architecture while replacing the learnable time domain encoder-decoder with STFT and iSTFT blocks to better address high reverberation scenarios. Concurrently, we estimate the activity patterns of the separated speech signal. The resulting voice activity detector (VAD) decisions can be further used in downstream tasks.

4.1 Problem Formulation

Let $x(t)$ be a mixture of I concurrent speakers captured by a single microphone:

$$x(t) = \sum_{i=1}^{I} \{s_i * h_i\}(t) + n(t) \quad t = 0, 1, \ldots, T-1, \tag{4.1}$$

where $s_i(t)$ represents the signal of the i-th speaker, $h_i(t)$ represents the, possibly time-varying, RIR between the i-th speaker and the microphone, and $n(t)$ represents an additive noise. Time variations of the RIR can be attributed to either the movements of the sources or the microphone, or both. In the STFT domain, and under the assumption of sufficiently long time frames, (4.1) can be formulated as,

$$x(l,k) = \sum_{i=1}^{I} s_i(l,k) h_i(l,k) + n(l,k), \tag{4.2}$$

where $l \in \{0, \ldots, L-1\}$ and $k \in \{0, \ldots, K-1\}$ are the time-frame and the frequency-bin (TF) indexes, respectively, with L the total number of time-frames and K the total number of frequency bands. In our study, we only address the case $I = 2$. Denote the outputs of the separator system in the STFT domain as $\hat{s}_1(l,k)$ and $\hat{s}_2(l,k)$.

We assume a fully-supervised setting, in which the goal is to infer a model that separates two output signals $\hat{s}_1(t)$ and $\hat{s}_2(t)$ from an unseen mixture $x(t)$, by maximizing the SI-SDR between the separated signals and the reverberated clean signals, $\{s_1 * h_1\}(t)$ and $\{s_2 * h_2\}(t)$, respectively.

4.2 Proposed Model

The model comprises two main components, namely a separation module and a VAD module. The separation module is based on a temporal convolutional network (TCN) backbone [13], similar to the Conv-Tasnet architecture [15]. Rather than a learned encoder and decoder, we use the STFT and the iSTFT. As demonstrated in [4, 25], audio processing algorithms that are based on STFT-iSTFT are advantageous in high reverberation as compared with learned encoder-decoder.

The VAD module determines the activity patterns of each of the separated signals and can be useful in downstream tasks. A block diagram of the entire system is depicted in Fig. 4.1. Next, we give more details on the various components of the system.

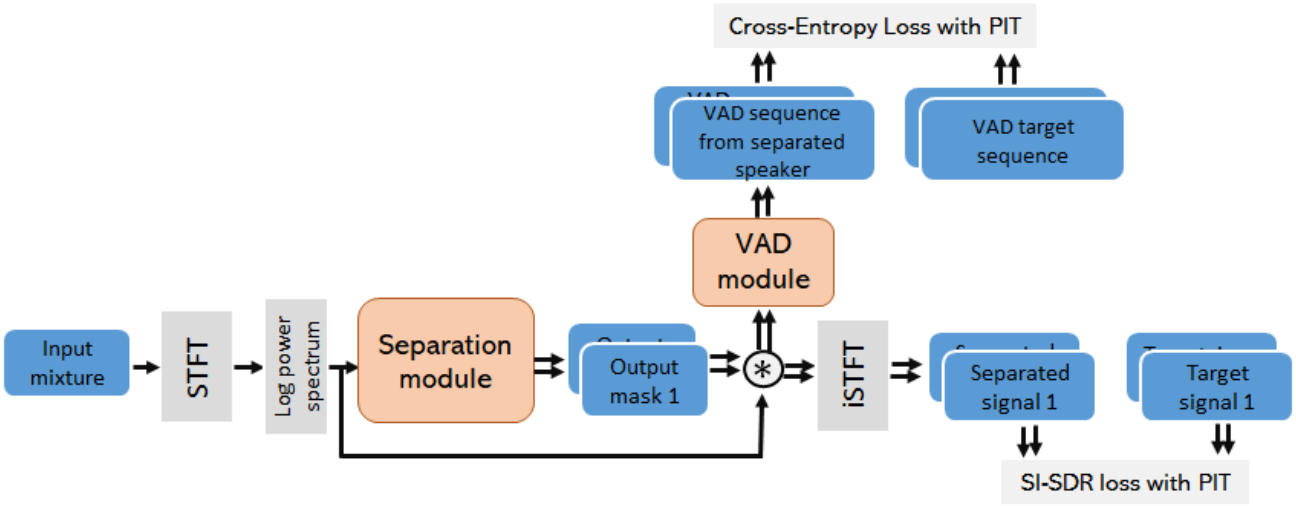


Figure 4.1: Audio separation and VAD network architecture. Learnable blocks are depicted in orange and data blocks are in blue.

4.2.1 Separation Module

The Separation Module is the main component of our proposed scheme. The main blocks of this module are depicted in Fig. 4.2. The raw audio signal is first analyzed by an STFT. The log-spectrum representation constitutes the input to the separation network after layer normalization.

The main processing module is an adapted TCN (dashed part in Fig. 4.2a). Originally, the TCN is a series of identical 1-D Conv blocks with increasing dilation factors, and with zero-padding along the time dimension to maintain their integrity. The dilation factor enables the capturing of a sufficiently long temporal context of the speech signal. Note that in high reverberation levels, past STFT frames are also relevant for the separation task and should be considered. Here, the adapted TCN consists of three repeats of a stack of eight 1-D AttConv blocks, as proposed in [15] and depicted in Fig. 4.2a, 4.2b.

Unlike the original 1-D Conv blocks in the Conv-Tasnet algorithm [15], our 1-D AttConv blocks have only one output and do not feature an additional skip connection output, from each block output to the overall TCN output (see Fig. 4.2b). We have decided to avoid this additional output since according to empirical evidence indicated in [19] and affirmed by our findings, they do not improve performance but rather increase the number of parameters.

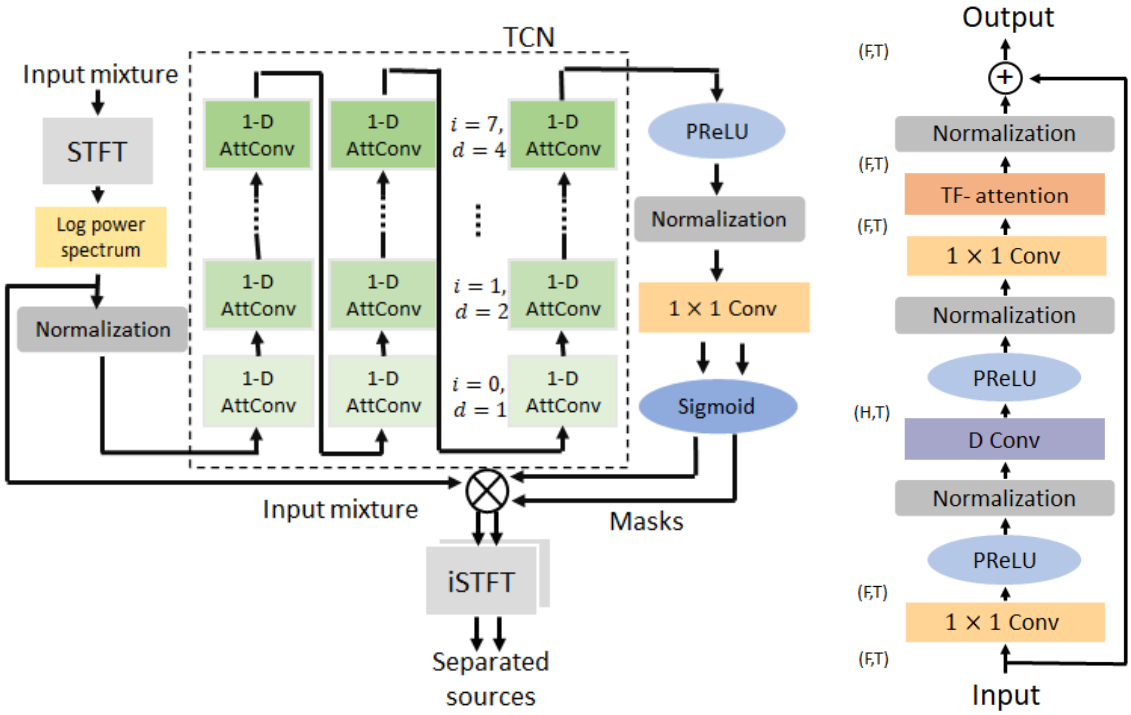
For each 1-D AttConv block (see Fig. 4.2b), the input of the block is summed to its output. Additionally, since the input to the network is an STFT representation, the dilation factors were chosen to maintain a small receptive field, $d = (i \bmod 4) + 1,\ 0 \leq i \leq 7$, where d is the dilation factor and i denotes the number of the 1-D AttConv block in each repeat as depicted in Fig. 4.2b.

The first 1×1 Conv layer in each 1-D AttConv block has a kernel size of 1. The number of filters is set to F, the number of frequency bins, to capture the frame-wise frequency patterns. In addition, following the 1×1 Conv layer, the D-Conv layer expounded upon in [15], is subsequently applied with H output filters, with the purpose of achieving a richer representation while concurrently minimizing the number of parameters.

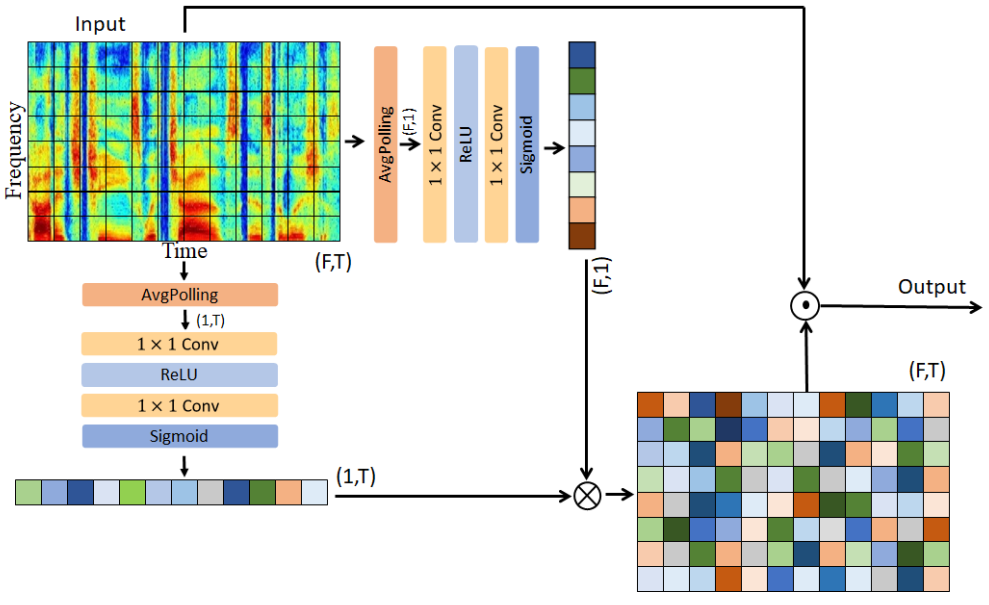
Another component contributing to the network's performance is the time-frequency attention block which is adopted from [31], where it was proposed in the context of noise reduction. The TF attention block is depicted in Fig. 4.2c. This block is positioned after the final 1×1 Conv layer of each 1-D AttConv block, followed by a normalization layer, to further optimize the network's ability to learn to recognize complex patterns in speech data. Despite its relatively small number of parameters, this module improves the performance by approximately 0.5 dB. As depicted in Fig. 4.2c, the block commences with average pooling layers on both the time and frequency dimensions, which are subsequently followed by 1×1 Conv layers and activation functions. These layers generate an attention mask that is then multiplied element-wise with the input spectrogram.

Finally, subsequent to the application of all the 1-D AttConv blocks, a pre-exponential linear unit (PReLU) activation function, along with a layer normalization and a 1×1 Conv, is employed to estimate the masks, which are subsequently passed through a Sigmoid activation function, such that its output is confined to the interval $[0, 1]$ as depicted in Fig. 4.2b. The estimated speakers' signals are transformed back to the time domain by augmenting the masked spectrum with the phase of the noisy and mixed input and then applying the iSTFT.

To facilitate low-latency implementation, we propose to apply the algorithm to 3 Sec long signals with a 1 Sec look ahead.



(a) Separation module (b) 1-D AttConv



(c) TF attention layer

Figure 4.2: Separation Module: Architecture.

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

4.2.2 VAD network

The purpose of the VAD network is to infer the activity patterns of the separated speakers. The inputs to the network are the extracted masks. The VAD network consists of a 1-D convolution layer with four filters, followed by a parametric Relu activation function and a normalization layer. The activity patterns of the corresponding speakers are finally obtained by applying a 1-D convolutional layer with one filter, followed by a hard threshold to obtain a binary activity decision for each frame.

The VAD decisions can be used for multiple downstream tasks, e.g., 1) a post-filter to further suppress residual interfering signals and noise, 2) indicators for the diarisation task to be used by the dialogue system, and 3) to facilitate parameter estimation of subsequent multichannel processing (see, e.g., the linearly constrained minimum variance (LCMV) beamformer discussed in D3.3).

4.2.3 Objective Functions

In order to efficiently train the model we experimented with several objective functions and found that the SI-SDR loss yields the best perceptual improvement. To alleviate the permutation problem, common to separation problems, uPIT [12] was employed. We stress that the target signals during training were the reverberant signals, namely the anechoic signals convolved with the corresponding RIRs. Hence, the signal focuses on the separation task and does not attempt to dereverberate the separated signals. While this may improve separation scores, in high reverberation levels the performance of ASR systems may deteriorate.

4.2.4 Training Procedure

A learning rate of $1e^{-3}$ was chosen, coupled with the Adam optimizer. After observing a slight difference between the validation and train scores, we carried out an experiment to evaluate the stochastic weight averaging (SWA) method. This approach involves the computation of the average of the weights resulting from the stochastic gradient descent (SGD) procedure, using a modified learning rate schedule. However, we did not observe any substantial improvement with this method. Furthermore, modifying the regularization parameter did not mitigate the minor discrepancy between the validation and train scores. The model was trained on our GPU servers, Tesla V100 SXM2 32GB, using 220 epochs. The parameter of the STFT and hyperparameters of the network are shown in Table 4.1

Table 4.1: Network's hyperparameters

	Variable	Value
STFT	Hop length	256
	FFT bins	512
	Window	Hamming
	Window length	512
Hyperparameter	H	512
	F	256
	F_s	16 kHz
	Batch size	16

4.3 Experimental Study

4.3.1 Databases

The WHAMR! dataset [17] is a widely-used database for speech separation in reverberant environments. Unfortunately, the reported reverberation level, $0.1 - 1$ s, is inaccurate, and in practice, it is much lower.

We have therefore constructed our own database to further evaluate the performance of the proposed algorithm. Clean speech signals were drawn from the Librispeech database and then convolved with RIRs generated using the image method [9]. The reverberation level is uniformly set in the range $[0.2, 0.6]$ s. The reverberated signals were then mixed with SIR=0 dB and contaminated with babble noise from the WHAM! dataset [26] with SNR uniformly drawn in

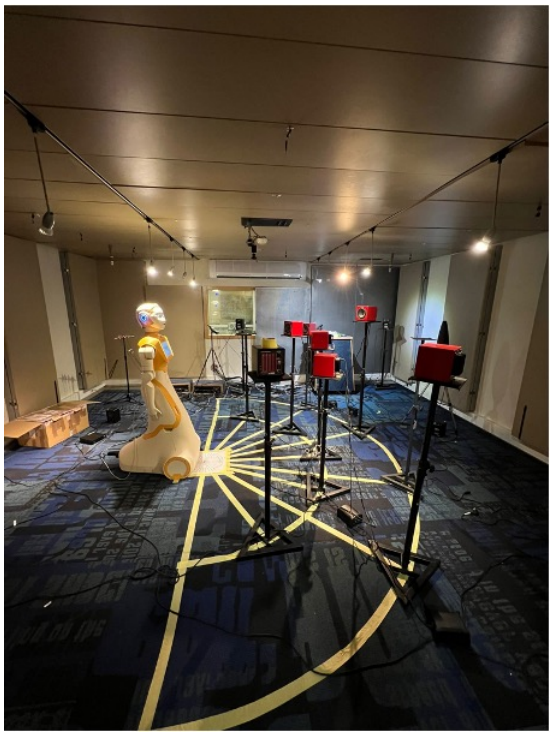


Figure 4.3: Recording setup with ARI (BIU Acoustic Lab).

the range $[0, 15]$ dB. The speakers were located in random positions (under physical constraints) inside shoebox rooms with random dimensions. Room dimensions were set in the range $H_x, H_y = U[4.5, 6.5]$ m, and $H_z = U[2.5, 3]$ m. The speakers are partially overlapping, with overlap percentage randomly selected from $[50\%, 75\%, 100\%]$. The length of each sample is 10 Sec. Overall, our simulated data consists of 155 hours of audio recordings for train, 39 for validation, and 22 for testing.

We also recorded 200 mixed signals using ARI's microphone array (only one channel is used for this algorithm). ARI was positioned at the center of our acoustic lab, with dimensions $[6, 6, 2.4]$ m, set to a reverberation level of $T_{60} = 350$ ms. The overlap between the speakers was randomly set in the range $[0.25\%, 0.5\%]$. The positions of the two speakers were randomly selected from the set of angles $[-65, -30, 0, 30, 65]^\circ$, 1 m from the robot. No external noise was added to the recordings (so only sensor noise and low-level ambient noise are present in the recordings). The lab setup is depicted in Fig. 4.3.

In addition, we simulated 200 samples of dynamic scenarios in order to examine the performance of the model in dynamic environments using the signal generator package.[1] Similar to the experimental conditions in the lab, the room was $[6 \times 6 \times 3]$ m, the reverberation level was set to $T_{60} = 350$ ms, with babble noise at $SNR = 25$ dB and the overlap was randomly set in the range $[0.25\%, 0.5\%]$. The cardioid microphone moves in a circular trajectory with a radius of 0.3 m and a velocity of 2.5 m/s. The static speakers were located 1 m from the center of the microphone trajectory at a random fixed angle.

4.3.2 Baseline Methods

The proposed model was compared with two SOTA methods, the SuDoRmRf [21, 22]. This model leverages the effectiveness of iterative temporal resampling strategies to avoid the need for multiple stacked dilated convolutional layers. It is important to note, however, that the receptive field of this network is quite large. Therefore, if one intends to use this network on a short buffer, it may pose a challenge.

Another baseline method is the Conv-Tasnet algorithm [15], as it shares a similar backbone with our separation module, namely the TCN module. There are several critical differences between our approach and the original Conv-Tasnet method, as explained above. A critical limitation of this baseline is its high memory consumption which may pose challenges for low-resource edge devices.

[1] Available online at github.com/ehabets/Signal-Generator

4.3.3 Results: Simulated Data

The average SI-SDR at the output of the proposed algorithm as compared with the competing algorithms for both the WHAMR! database and the new simulated data are depicted in Table 4.2.

Table 4.2: Mean separation results in SI-SDR [dB]

Algorithm	WHAMR!	Simulated Data
Proposed	8.1	6.92
SuDoRmRf	7.04	5.6
Conv-Tasnet	8.3	6.6

The SI-SDR measures of the proposed method and the Conv-Tasnet are comparable. Recall that the memory consumption of the proposed method is significantly lower. It is also evident that the new simulated data is more challenging than the WHAMR! database.

A more detailed analysis of the SI-SDR improvement of the proposed method is depicted in Fig. 4.4. The histograms clearly indicate a significant improvement (almost 9 dB in the mean score). Moreover, the long histogram tail indicates that for a substantial number of utterances, the proposed method fails to improve the SI-SDR. Further insights on

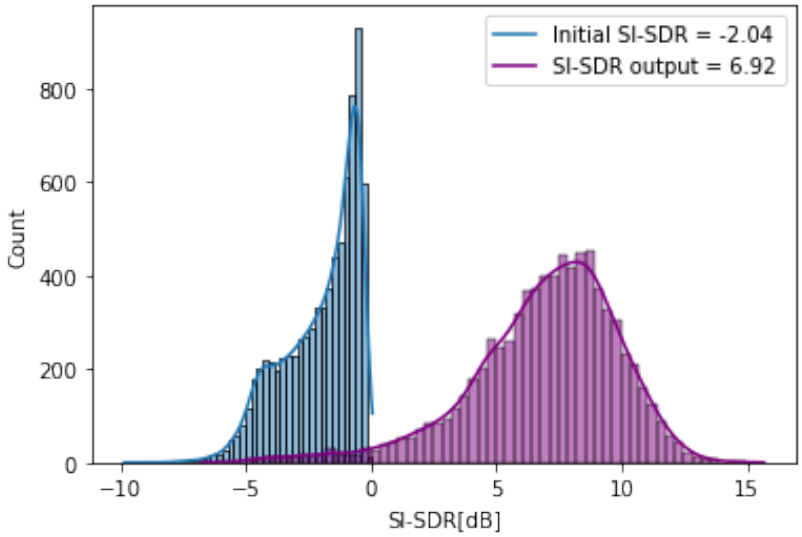
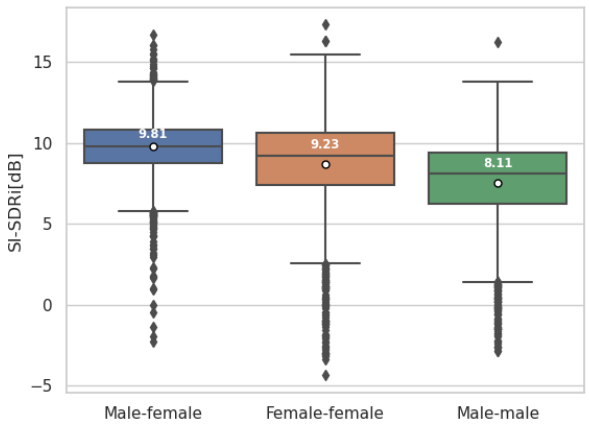


Figure 4.4: Histogram of the SI-SDR at the input and the output of the proposed algorithm for the simulated data.

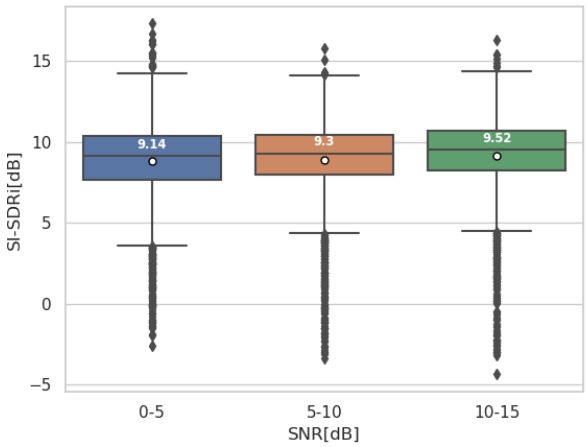
the performance of the proposed algorithm can be inferred from the following box plots. Figure 4.5, depicts the SI-SDR improvement categorized according to gender, signal-to-noise ratio (SNR) level, reverberation level, and direct-to-reverberation ratio (DRR). It is evident from Fig. 4.5a that the best results are obtained for the mixed-gender case, while the worst results are for the male-male mixture. A marginal improvement is demonstrated with increasing SNR level, as evident from Fig.4.5b. The dependency on the reverberation level is indicated in Fig. 4.5c, with a clear performance drop from lower levels to higher levels of T_{60}. Finally, as depicted in Fig. 4.5d, best results are obtained if the distance of both sources from the microphone is lower than the critical distance (and consequently high DRR), and worst when both sources are beyond the critical distance. It should be stressed that above the critical distance, the reverberation tail dominates the RIR and hence the signal is perceived as more reverberant.

4.3.4 Results: Recorded Data on Static ARI

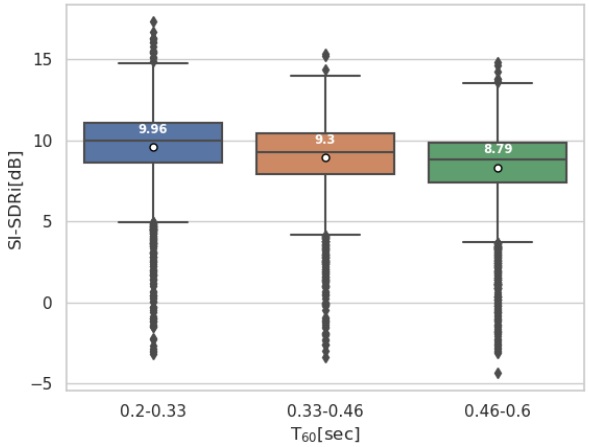
We have also used the English database recorded on ARI to verify the applicability of the algorithm to our scenarios. A significant improvement both in terms of WER (using NVIDIA's ASR) and SI-SDR can be clearly deduced from the histograms in Fig. 4.6. The median SI-SDR was improved from approximately 0 dB to 11.74 dB, and the median WER from 76% to 21%. Note that in this experiment we have used noiseless recordings.



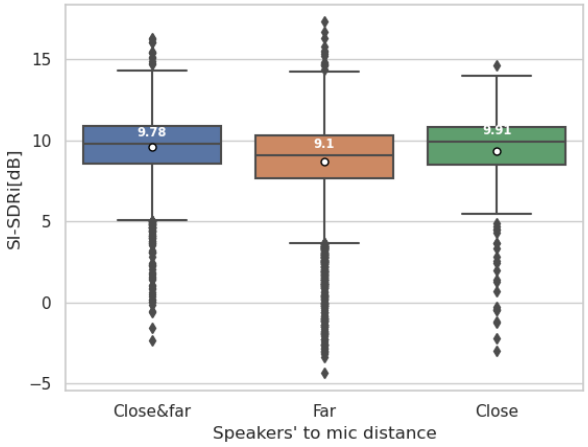
(a) scale-invariant signal-to-distortion ratio improvment (SI-SDRi) vs. Gender



(b) SI-SDRi vs. SNR

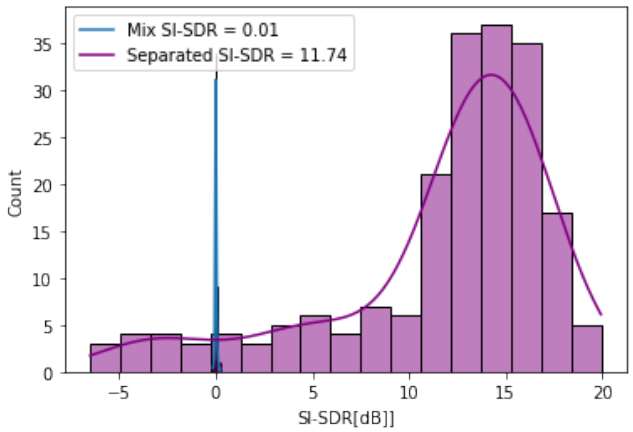


(c) SI-SDRi vs. T_{60}

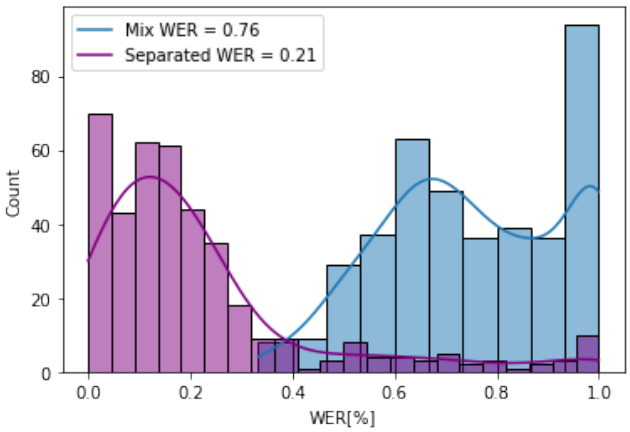


(d) SI-SDRi vs. speakers' to mic distance

Figure 4.5: SI-SDRi for various dependencies. The number inside the boxplot is the median and the point indicates the mean score



(a) SI-SDR histograms

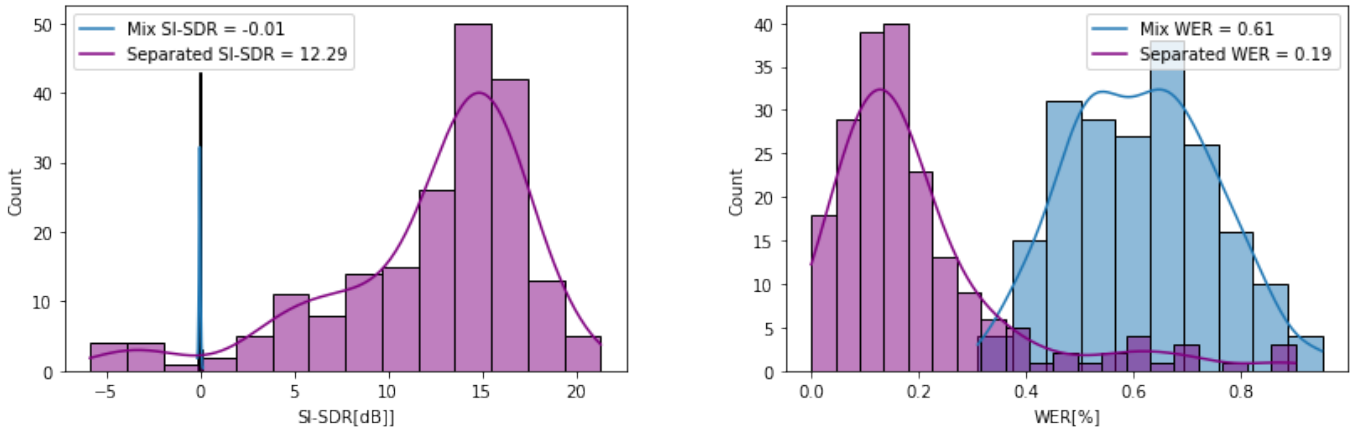


(b) WER histograms

Figure 4.6: SI-SDR and WER improvements with ARI recordings, $T_{60} = 0.35$ s and low sensor noise. The overlap of the speakers was randomly set to the range $[0.25\%, 0.5\%]$.

4.3.5 Results: Simulated Dynamic Microphone Data

We tested our network on the simulated dynamic microphone data. A significant improvement both in terms of WER (using NVIDIA's ASR) and SI-SDR can be clearly observed by inspecting the histograms in Fig. 4.7. The mean SI-SDR was improved from approximately 0 dB to 12.29 dB, and the mean WER from 61% to 19%. Moreover, we can clearly observe a shift of the histogram from high WER levels to low levels.



(a) SI-SDR histograms

(b) WER histograms

Figure 4.7: SI-SDR and WER improvements with simulated dyanmic microphone data, $T_{60} = 0.35$ s and with $SNR = 25$ dB. The overlap of the speakers was randomly set to the range $[0.25\%, 0.5\%]$.

4.3.6 Results: Recorded Dynamic Microphone Data

The ultimate goal of the audio separation algorithm is to improve ASR results while ARI is moving. When ARI is moving it generates noise (due to the friction of the wheels with the floor and the joint movement). We are currently working on suppressing the so-called, ego noise and will report on this algorithm elsewhere. Meanwhile, to imitate a moving microphone, we followed another procedure.

Rather than using the microphones(s) that are mounted inside ARI, we have used an external ReSpeaker microphone array (identical to the device installed in ARI). The user was holding the device in his hands while slowly walking in circles. During the array movement, speech utterances were played from two static loudspeakers.

As demonstrated in Table 4.3, preliminary results are very encouraging. However, we note that ARI's ego noise may deteriorate the results.

Table 4.3: Sample transcription of a sound mixture captured by a moving ReSpeaker array.

	True Transcription	ASR Transcription	WER
Mix	-	they're kind of discipline whether addressed to her mind or heart little coral might or might not be within its reach in accordance with just finished with a spasm down his strained gullet when the baffled hawk caught sight of him and swooped yes	70%, 78%
Speaker #1	OTHER kind of discipline whether addressed to her mind or heart little PEARL might or might not be within its reach in accordance with the CAPRICE that RULED the moment	THEY'RE kind of discipline whether addressed to her mind or heart little CORAL might or might not be within its reach in accordance with the PRIEST that ROLE AT the moment	16%
Speaker #2	in FACT he had just finished it the LAST of the TROUT'S tail had just vanished with a spasm down his strained gullet when the baffled hawk caught sight of him and swooped	in FA he had just finished it the LAUGH of the CHILD'S tail had just vanished with a spasm down his strained gullet when the baffled hawk caught sight of him and swooped YES	12%

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

5 Conclusions

This document reports the progress in the speaker separation task. We presented two new single-microphone algorithms, developed in the course of the SPRING project. The first is a speaker extraction algorithm, which can be used when a reference utterance of the desired speaker is available. This may be obtained by identifying time frames in the same conversation (about 1 Sec long) in which only a single speaker is active, e.g. by utilizing the multi-channel current speakers activity detector (MCCSD) [1]. The second is a speaker separation algorithm that employs TCN module. Both algorithms are implemented in Python. The second is already deployed on ARI as part of the audio pipeline.

The extraction algorithm obtains excellent improvements in both SI-SDR and STOI measures for a medium-high reverberation level. The separation algorithm was tested with NVIDIA's ASR system and provided significant improvements in medium reverberation levels, $T_{60} \approx 350$ ms, and low noise levels. The speaker overlap was randomly chosen between 25% and 50%, as scenarios with full overlap between speakers are not assumed realistic. We also tested the algorithm with a *moving* ReSpeaker sound card, which is not embedded in ARI. Again, significant ASR improvements are demonstrated in a low-noise environment.

We note that in the current audio processing architecture, two independent audio streams are simultaneously transcribed by the ASR system and transmitted to the dialogue system. To circumvent the speaker permutation phenomenon, typical to separation algorithms, we will apply a speaker identification module on the separated outputs and preserve the time consistency of the transcribed speech signals. For the extraction algorithm, we may also use the identity determined by the reference speaker, and for the separation algorithm the VAD decisions.

The operation paradigm of the dialogue system with two concurrent transcriptions should still be determined.

The final decision on which algorithm will be eventually deployed on ARI will be made at a later stage, based on the following considerations: 1) performance (in terms of ASR accuracy) in the target acoustic environment; 2) robustness to changing conditions; 3) computational load and memory requirements; and 4) system perspective, e.g. how well the algorithms fits the entire audio pipeline, especially if several enhancement algorithms should be cascaded.

Finalizing the entire audio pipeline is one of the significant remaining challenges. We already know that ARI self-noise deteriorates the ASR results and the separation capabilities. We are currently examining a cascade of a narrow-band noise reduction algorithm [29] followed by one of the separation algorithms.

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

Bibliography

[1] Shlomo E. Chazan, Jacob Goldberger, and Sharon Gannot. LCMV beamformer with DNN-based multichannel concurrent speakers detector. In *The 26th European Signal Processing Conference (EUSIPCO)*, Rome, Italy, September 2018.

[2] Shlomo E Chazan, Lior Wolf, Eliya Nachmani, and Yossi Adi. Single channel voice separation for unknown number of speakers under reverberant and noisy settings. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3730–3734, 2021.

[3] Jingjing Chen, Qirong Mao, and Dong Liu. Dual-Path Transformer Network: Direct Context-Aware Modeling for End-to-End Monaural Speech Separation. In *Proc. Interspeech*, pages 2642–2646, 2020.

[4] Tobias Cord-Landwehr, Christoph Boeddeker, Thilo Von Neumann, Cătălin Zorilă, Rama Doddipatla, and Reinhold Haeb-Umbach. Monaural source separation: From anechoic to reverberant environments. In *International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2022.

[5] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2010.

[6] Marc Delcroix, Tsubasa Ochiai, Katerina Žmolíková, Kateřina, Keisuke Kinoshita, Naohiro Tawara, Tomohiro Nakatani, and Shoko Araki. Improving speaker discrimination of target speech extraction with time-domain speakerbeam. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 691–695, 2020.

[7] Masood Delfarah, Yuzhou Liu, and DeLiang Wang. A two-stage deep learning algorithm for talker-independent speaker separation in reverberant conditions. *The Journal of the Acoustical Society of America*, 148(3):1157–1168, 2020.

[8] Aviad Eisenberg, Sharon Gannot, and Shlomo E Chazan. Single microphone speaker extraction using unified time-frequency siamese-unet. In *30th European Signal Processing Conference (EUSIPCO)*, pages 762–766, 2022.

[9] Emanuël AP Habets. Room impulse response generator. Technical report, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2014.

[10] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 31–35, 2016.

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

[12] Morten Kolbæk, Dong Yu, Zheng-Hua Tan, and Jesper Jensen. Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(10):1901–1913, 2017.

[13] Colin Lea, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks: A unified approach to action segmentation. In *European Conference on Computer Vision (ECCV)*, pages 47–54, Amsterdam, The Netherlands, October 2016. Springer.

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

[15] Yi Luo and Nima Mesgarani. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM transactions on audio, speech, and language processing*, 27(8):1256–1266, 2019.

[16] Shahar Lutati, Eliya Nachmani, and Lior Wolf. SepIt approaching a single channel speech separation bound. *arXiv preprint arXiv:2205.11801*, 2022.

[17] Matthew Maciejewski, Gordon Wichern, Emmett McQuinn, and Jonathan Le Roux. WHAMR!: Noisy and reverberant single-channel speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 696–700, 2020.

[18] Eliya Nachmani, Yossi Adi, and Lior Wolf. Voice separation with an unknown number of multiple speakers. In *International Conference on Machine Learning (ICML)*, pages 7164–7175, 2020.

[19] William Ravenscroft, Stefan Goetze, and Thomas Hain. Deformable temporal convolutional networks for monaural noisy reverberant speech separation. *arXiv preprint arXiv:2210.15305*, 2022.

[20] Cem Subakan, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong. Attention is all you need in speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 21–25, 2021.

[21] Efthymios Tzinis, Zhepei Wang, Xilin Jiang, and Paris Smaragdis. Compute and memory efficient universal sound source separation. *Journal of Signal Processing Systems*, 94(2):245–259, 2022.

[22] Efthymios Tzinis, Zhepei Wang, and Paris Smaragdis. Sudo rm-rf: Efficient networks for universal audio source separation. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2020.

[23] Kai Wang, Hao Huang, Ying Hu, Zhihua Huang, and Sheng Li. End-to-end speech separation using orthogonal representation in complex and real time-frequency domain. In *Interspeech*, pages 3046–3050, 2021.

[24] Zhong-Qiu Wang, Gordon Wichern, and Jonathan Le Roux. Convolutive prediction for monaural speech dereverberation and noisy-reverberant speaker separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3476–3490, 2021.

[25] Zhong-Qiu Wang, Gordon Wichern, Shinji Watanabe, and Jonathan Le Roux. STFT-domain neural speech enhancement with very low algorithmic latency. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:397–410, 2022.

[26] Gordon Wichern, Joe Antognini, Michael Flynn, Licheng Richard Zhu, Emmett McQuinn, Dwight Crow, Ethan Manilow, and Jonathan Le Roux. Wham!: Extending speech separation to noisy environments. *arXiv preprint arXiv:1907.01160*, 2019.

[27] Chenglin Xu, Wei Rao, Eng Siong Chng, and Haizhou Li. Time-domain speaker extraction network. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 327–334, 2019.

[28] Chenglin Xu, Wei Rao, Eng Siong Chng, and Haizhou Li. Spex: Multi-scale time domain speaker extraction network. *IEEE/ACM transactions on audio, speech, and language processing*, 28:1370–1384, 2020.

[29] Yochai Yemini, Shlomo E Chazan, Jacob Goldberger, and Sharon Gannot. A composite DNN architecture for speech enhancement. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 841–845, 2020.

[30] Neil Zeghidour and David Grangier. Wavesplit: End-to-end speech separation by speaker clustering. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:2840–2849, 2021.

[31] Qiquan Zhang, Xinyuan Qian, Zhaoheng Ni, Aaron Nicolson, Eliathamby Ambikairajah, and Haizhou Li. A time-frequency attention module for neural speech enhancement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:462–475, 2023.

[32] Kateřina Žmolíková, Marc Delcroix, Keisuke Kinoshita, Tsubasa Ochiai, Tomohiro Nakatani, Lukáš Burget, and Jan Černockỳ. Speakerbeam: Speaker aware neural network for target speaker extraction in speech mixtures. *IEEE Journal of Selected Topics in Signal Processing*, 13(4):800–814, 2019.