


EMPIRICAL RESEARCH

Open Access



# Robustness of ad hoc microphone clustering using speaker embeddings: evaluation under realistic and challenging scenarios

Stijn Kindt<sup>1\*</sup> , Jenthe Thienpondt<sup>1</sup>, Luca Becker<sup>2</sup> and Nilesh Madhu<sup>1</sup>

## Abstract

Speaker embeddings, from the ECAPA-TDNN speaker verification network, were recently introduced as features for the task of clustering microphones in ad hoc arrays. Our previous work demonstrated that, in comparison to signal-based Mod-MFCC features, using speaker embeddings yielded a more robust and logical clustering of the microphones around the sources of interest. This work aims to further establish speaker embeddings as a robust feature for ad hoc microphone clustering by addressing open and additional questions of practical interest, arising from our prior work. Specifically, whereas our initial work made use of simulated data based on shoe-box acoustics models, we now present a more thorough analysis in more realistic settings. Furthermore, we investigate additional important considerations such as the choice of the distance metric used in the fuzzy C-means clustering; the minimal time range across which data need to be aggregated to obtain robust clusters; and the performance of the features in increasingly more challenging situations, and with multiple speakers. We also contrast the results on the basis of several metrics for quantifying the quality of such ad hoc clusters. Results indicate that the speaker embeddings are robust to short inference times, and deliver logical and useful clusters, even when the sources are very close to each other.

**Keywords** Acoustic sensor networks (ASN), Distributed microphone clustering, Microphone clustering metrics, Ad hoc speaker separation

## 1 Introduction

Many ‘smart’ devices carry at least one microphone. Typical examples are phones, smart watches and laptops. There is also a trend towards the internet of things (IoT) and smart homes, increasing the number of microphone-carrying devices scattered around a room. Sharing information from all these microphones, by forming an acoustic sensor network (ASN), can give a good acoustic

coverage of a room/living environment. This can be exploited for tasks like acoustic event detection, classification, and separation, in scenarios such as assisted living and healthcare, hearing aids, and communications (see, e.g. [1]).

Since the microphones can be distributed all over the room, the spatial diversity is greater than that of a compact microphone array (microphones in close proximity). However, combining the signals of such distributed microphones is not straightforward. Firstly, the microphones may not be driven by the same clock, so sample rate offsets (SROs) and sample time offsets (STOs) may be present. The relative time delay between signals at different microphones is therefore no longer only an effect of the propagation delays. Additionally, if the ASN is connected via wireless links (WASNs), bandwidth and

\*Correspondence:

Stijn Kindt  
[stijn.kindt@ugent.be](mailto:stijn.kindt@ugent.be)

<sup>1</sup> IDLab, Department of Electronics and Information Systems, Ghent University - imec, Ghent, Belgium

<sup>2</sup> Institute of Communication Acoustics, Ruhr-Universität Bochum, Bochum, Germany

processing power limitations are introduced. Furthermore, for portable microphone-carrying devices, the position of the microphones is not known a priori, and forming an ASN from such ad hoc distributed microphones makes it even harder to perform localisation or separation.

In order to cope with the unknown microphone positions, it is often helpful to cluster microphones based on the similarity of the signals they capture. Thereby all microphones dominated by the same source may be expected to be grouped in the same cluster. Similar clustering of microphones which primarily pick up the ambient signal or noise can be performed. Such clustering has already been proven valuable for subsequent steps like source classification (e.g. [2, 3]) and separation (e.g. [4, 5]).

The clustering procedure consists of two main stages: (i) proper selection of acoustic features, upon which clustering is carried out, and (ii) choosing an appropriate clustering algorithm. Below we first discuss prior work in this regard, before outlining the main contributions of our work.

### 1.1 Prior work

A variety of clustering features have been proposed in the literature. For example, the magnitude squared coherence (MSC) between microphones on the noise-only part of the signal is used in [4]. Assuming the noise field to be diffuse gives a direct relation between the noise-MSC and the inter-microphone distance. In a similar vein, the room impulse responses (RIRs) are first estimated for each microphone in [6], and are subsequently used to cluster microphones. Such classes of techniques depend solely on the room properties to perform clustering.

In contrast, the MSC on the *speech-active* parts of the signal is utilised as cluster features in [3]. This contains information about the RIRs and the *content* of the signal, thus both the room characteristics and signal correlation is exploited. Similarly, in [7], the individual microphone auto-correlation of the source signal and the auto-correlation of the noise signal are computed, where identification of the noise and source regions is done with the help of voice activity detection (VAD). These yield source- and location-specific features, which are used for the clustering.

All the above-mentioned techniques are influenced by the room characteristics. These characteristics could be useful if geometry-related information is required, e.g. to estimate the position of the microphones in the room. This would however also require a prior calibration stage for different positions in the room, as done in [8]. In contrast, features that are *speech- or content-specific* are useful to be able to focus on

pre-determined targets (e.g. in care homes, where monitoring of particular patients may be desired). Additionally, *speaker-specific* features can lead to a more targeted clustering, and without the need to first estimate the room-acoustics.

Clustering based on purely signal-dependent features has, therefore, also been investigated. The work in [2, 5, 9, 10] proposed hand-crafted features, based on the modulation-domain Mel frequency cepstral coefficients (Mod-MFCCs), where mean subtraction reduces the effect of the room characteristics under the assumption that the source and microphone stay sufficiently static. In contrast, the work in [11] depends on data-driven feature extraction, where a variational auto-encoder (VAE) trained on all types of speech and music data is used within a federated learning framework. After training, the parameters of the bottleneck layer are randomised and the model is distributed to all the microphone nodes. During runtime, each node updates the bottleneck weights based on the captured signal, essentially overfitting on that signal. The accumulated gradients from multiple rounds of backpropagation are sent back to the central node and are used as cluster features. The advantage of this approach is the privacy preservation of the speaker. However, the privacy constraint inevitably precludes the use case where speaker-specific processing is desired. Also, retraining the network at each node comes at a relatively high computational cost, which has been discussed and improved in [12].

Since the primary goal is to detect and cluster microphones around speech sources, we introduced speaker embeddings — representation of a talker in a high-dimensional *latent space* — as features in [13]. The embeddings are generated by a pre-trained speaker verification network: the Enhanced Propagation and Aggregation Time Delay Neural Network (ECAPA-TDNN) [14]. Since speaker verification should be robust to different room characteristics and perturbations, the embedding network is trained with appropriately augmented data, yielding room-independent and yet source-specific embeddings, which serve well as clustering features.

For the clustering algorithm itself, we note that there are many approaches in the literature, e.g. K-means is used in [6], non-negative matrix factorisation is utilised in [3], while matrix bi-partitioning is deployed in [11]. In contrast, fuzzy C-means (FCM) is incorporated in approaches based on the mod-MFCC features [2, 5, 10]. The fuzzy weights indicate the *degree* to which a microphone belongs to a cluster — which is indirectly an indication of the strength of the target source at that microphone. Therefore, we also adopted FCM in our approach, as the fuzzy weights can be informative for later stages, like enhancing the source.

This work builds on the initial results of [13]. The goal is to obtain a holistic overview of the opportunities and limitations of using speaker embeddings as clustering features for ad hoc distributed microphones. The main contributions of this work are outlined below.

## 1.2 Contributions

For the FCM clustering, the standard Euclidean distance was used in previous work, whereas speaker verification implementations typically use the cosine similarity, as the direction and orientation of the embeddings yield more discrimination. Therefore, as part of this work, we investigate the benefit of using the cosine distance in FCM-based clustering.

Also, our initial comparison [13] of the speaker embeddings with the Mod-MFCC-based features was in simulated shoe-box rooms. There it was shown that the embeddings generate more robust and visually logical clusters. It was also assumed that the sources to be separated were sufficiently far apart and the feature extraction was on data aggregated across a relatively long time-span of 4 s. This initial study raised several interesting questions, which are handled in the current contribution, namely: (i) what is the effect of realistic room environments on the features? (ii) As mentioned above, what distance metric is best suited for the clustering? (iii) What happens if the sources were placed in close proximity? (iv) Does the time-scale of data aggregation affect the performance? And, last but not least, (v) can speaker-embedding-based features be used to detect the presence of known talkers and only extract them in realistic, *dialogue-like* situations? We believe that answering these questions is important to obtain a full picture for practical implementations.

For realistic room environments, we employ the SINS database [15]. We systematically evaluate the performance on distant- and closely-spaced sources. Next, we vary the duration of the segment on which the clustering features are generated. The former will generate insights into the robustness of the features under increased difficulty, while the latter indicates the feasibility of adapting to quickly changing environments (more frequent updates on shorter segments) or of scaling the complexity (e.g. for bandwidth and power constraints) by updating less frequently and on shorter segments.

For quantitative appreciation of the results, a concept of cluster quality needs to be defined. However, this is not a trivial task, as generating the ground truth is not straightforward. Thus, we proposed three intuitive metrics in [13]: (I) the histograms of the direct-to-reverberant and (II) direct-to-reverberant-interference-and-noise ratios (DRR and DRINR) of microphones attributed to a speech-source cluster (indicating the quality of the

microphones allocated to a cluster), and (III) the average number of microphones in a speech-source cluster (indicating spatial diversity available at a cluster). Additionally, we also benchmark on cluster-based speaker separation from [5]. With these metrics taken together, we obtain a more holistic performance overview.

The rest of the paper is structured as follows: in Section 2, we will write out the signal model followed by a succinct explanation of the Mod-MFCC and speaker embedding-based features in Section 3. The FCM algorithm is discussed in Section 4, followed by the speaker separation scheme for evaluation in Section 5. Section 6 explains the different situations we evaluate, as well as the metrics we use to benchmark the clustering. The discussion of the results is done in Section 7, and Section 8 concludes the paper.

## 2 Signal model

For our setup, we consider  $J$  concurrently active sources and  $M$  microphones distributed in the room. The  $m$ th microphone signal,  $y_m$ , is given as:

$$y_m(n) = \sum_{j=1}^J x_{j,m}(n) + v_m(n), \quad (1)$$

where  $n$  is the discrete time index,  $x_{j,m}$  is the source signal captured by the  $m$ th microphone and generated by the  $j$ th source, and  $v_m$  symbolises the additive noise at the  $m$ th microphone.

In the following, we shall use the short-time Fourier transform (STFT) representation of the signal for processing. The signal in this domain is denoted as:

$$Y_m(l, k) = \text{STFT}[y_m(n)], \quad (2)$$

where  $k$  is the STFT frame index and  $l$  is the index of the discrete frequency bin.

## 3 Clustering features

As previously mentioned, there are three major categories of feature types on which clustering has been performed. The first set, based on estimating the relative locations of the microphones with respect to each other, is termed geometry-based features (GBFs). The second class of features exploits geometry and signal information and is termed as signal-based features (SBFs). The last set generates features that are source-specific and we term these source-dependent latent features (SDLFs).

GBFs extract information relating to the relative spatial distances between microphones. This can be obtained explicitly by estimating the RIRs ([6, 8]) or implicitly, using the coherence in the noise-only periods as in [4].

SBFs are computed by comparing signals across different microphones and typically contain information

on the acoustic environment and the source signals. The use of the MSC, as in [3, 7] are examples of such feature usage.

The use of SDLFs is based on the fundamental assumption that signals from microphones close to the same source will generate similar *latent* features. Additionally, if the latent features are designed to be source discriminating, features characterising one source should be very different from those for other sources and the ambient noise. A seminal example here is the set of hand-crafted Mod-MFCC features proposed in [9]. A data-driven approach to get SDLFs is proposed in [11], which is based on the use of auto-encoders and federated learning principles.

Although the latter two methods try to focus on the source-specific characteristics, there will always be some influence of the room characteristics — which reduces the discriminative capacity of these features. Therefore, we propose to use speaker verification networks to generate source-specific features, as these networks are trained to generate the same embedding for a speaker with relative robustness to the environmental conditions. Additionally, as the embeddings should be sufficiently unique in order to discriminate between *different* speakers, they can yield a robust indication of source dominance at a microphone — making them ideal for the application to ad hoc arrays.

Given our focus on demonstrating the benefits of source-specific features in ASNs, we limit ourselves to SDLFs in this study. Specifically, we use the Mod-MFCC features as a baseline for benchmarking speaker embedding features. The federated learning framework is not considered due to its large computational cost and complexity (multiple rounds of backpropagation are needed). Furthermore, in contrast to speaker embeddings, information about specific talkers cannot be exploited within this framework — making it less versatile.

### 3.1 MFCC-based features

The modulated Mel-frequency cepstral coefficients (Mod-MFCC) based features were first utilised in [2, 9]. These hand-engineered features consist of two  $\mathcal{N}$ -dimensional cepstral modulation ratios (CMR) and one  $\mathcal{N}$ -dimensional averaged modulation amplitude (AMA), where  $\mathcal{N}$  is the number of considered cepstrum bins.

We briefly summarise the computation of these features as proposed in [2] and subsequently denoted as  $\mathcal{F}^{MFCC}$ . First, the MFCC,  $Y_{MFCC}(\eta, k)$  are computed from the STFTs in (2). Here,  $\eta$  is the cepstral index. Cepstral mean subtraction (CMS) is applied to reduce the effect of reverberation, resulting in features that better capture the speech structure [16, 17].

$$\tilde{Y}_{MFCC}(\eta, k) = Y_{MFCC}(\eta, k) - \frac{1}{K} \sum_{k=0}^{K-1} Y_{MFCC}(\eta, k). \quad (3)$$

The Mod-MFCC is then calculated as the DFT of the MFCC features with a rectangular window of length  $L$ :

$$Y_{Mod-MFCC}(\kappa, \eta, \lambda) = \sum_{L=0}^{L-1} \tilde{Y}_{MFCC}(\eta, \lambda Q + l) e^{-j2\pi l \kappa / L}, \quad (4)$$

where  $\lambda \in \{0, \dots, \Lambda - 1\}$  is the modulation index,  $Q$  the modulation shift and  $\kappa \in \{0, \dots, L/2\}$  is the modulation frequency bin. Averaging the modulation amplitude spectra,  $|Y_{Mod-MFCC}(\kappa, \eta, \lambda)|$ , over time is done in order to be robust against time shifts that are expected in ASNs:

$$\hat{Y}_{Mod-MFCC}(\kappa, \eta) = \sum_{\lambda=0}^{\Lambda-1} |Y_{Mod-MFCC}(\kappa, \eta, \lambda)|. \quad (5)$$

Then the cepstral modulation ratio (CMR) features and averaged modulation amplitude (AMA) features are defined as:

$$CRM_{\kappa_1|\kappa_2}(\eta) = \frac{\sum_{\kappa=\kappa_1}^{\kappa_2} \hat{Y}_{Mod-MFCC}(\kappa, \eta)}{(\kappa_2 - \kappa_1 + 1) \hat{Y}_{Mod-MFCC}(0, \eta)}, \quad (6)$$

$$AMA(\eta) = \frac{1}{L/2 + 1} \sum_{\kappa=0}^{L/2} \hat{Y}_{Mod-MFCC}(\kappa, \eta). \quad (7)$$

The final MFCC-based feature vector is then:  $\mathcal{F}^{MFCC} = [AMA^T, \mathbf{CRM}_{1|1}^T, \mathbf{CRM}_{2|8}^T]^T$ , where  $\mathbf{AMA}$ ,  $\mathbf{CRM}_{1|1}$  and  $\mathbf{CRM}_{2|8}$  are  $\mathcal{N}$ -dimensional column vectors. The first cepstral bin is omitted ( $\eta \in \{1, \dots, \mathcal{N}\}$ ) to reduce sensitivity to the amplitude of the signals.

### 3.2 Speaker verification-based features

Speaker embeddings refer to the representation of a talker in a high-dimensional latent space. In speaker verification tasks, such embeddings are used to test if two audio utterances are spoken by the same person. For this, embeddings extracted from the utterances are compared using a similarity metric that is appropriate to the embedding extractor architecture. The utterances are accepted as coming from the same speaker if the similarity exceeds a predetermined threshold. Applied to our case, such embeddings, extracted from the individual microphone signals, can similarly be compared — whereby microphones dominated by the same speaker would yield embeddings that are near identical.

The embedding features are generated by the recent Emphasized Channel Attention, Propagation and



Aggregation Time Delay Neural Network (ECAPA-TDNN) [14]. ECAPA-TDNN improves upon the popular x-vector architecture [18] by introducing several enhancements. First, an attentive statistics pooling layer is incorporated into the network which emphasises important frame- and channel-level features during the statistics pooling operation. Additionally, a speech-adapted version of Squeeze-Excitation (SE) [19] is introduced to inject global context in the intermediate frame-level features of the model. Finally, multi-layer feature aggregation before the pooling layer gives the model the opportunity to incorporate information learned from multiple levels in the network. The ECAPA-TDNN model is optimised using the Additive Angular Margin (AAM) [20] softmax loss function. This enables us to also consider the cosine similarity as the similarity metric for comparing two embedding vectors. We use the same training procedure as described in [14].

The embedding features,  $\mathcal{F}^{SpVer}$ , extracted for each microphone, are directly input to the clustering algorithm.

#### 4 Fuzzy C-means clustering

We use, similar to Gergen et al. [2], the fuzzy C-means (FCM) algorithm to cluster the microphone features. FCM is closely related to the K-means algorithm, with the main difference being the fuzzy membership values (FMV) included in FCM. K-means generates hard clusters where a microphone is either part of the cluster or not. However, the FMVs, which reflect how much a microphone belongs to each cluster, are useful for subsequent processing. It can for instance be used to determine the *reference* microphone, or indicate that certain sources, although part of one cluster, also contains information about another cluster. The first is useful in estimating initial speaker separation masks [10], while the latter can reasonably increase the number of microphones to be included in beamforming efforts [5]. Additionally, the FMV can be used to inform a weighted delay-and-sum beamformer (DSB) [5]. These separation methods will be discussed in more detail in Section 5.

In general, we will generate  $C = J + 1$  fuzzy clusters. That is, one cluster for each source and one background (noise) cluster. The background cluster ideally collects all the microphones dominated by noise or reverberations, thus assuring that each microphone from a source cluster is dominated by that source.

##### 4.1 FCM algorithm

The FCM algorithm minimises the following weighed error function [21]:

$$\mathcal{L} = \sum_{c=0}^{C-1} \sum_{m=0}^{M-1} \mu_{m,c}^{\alpha} \delta(\mathcal{F}_m, \mathcal{C}_c) \quad (8)$$

where  $\mu_{m,c}$  are the FMVs,  $\delta(\mathcal{F}_m, \mathcal{C}_c)$  is the distance metric between the features of microphone  $m$  and the  $c$ th cluster centre  $\mathcal{C}_c$ , and  $\alpha$  is the fuzzy weighting exponent. Putting  $\alpha$  to 1 will result in hard clusters, while setting  $\alpha \rightarrow \infty$  will result in  $\mu_{m,c} \rightarrow 1/C$ ; thus, a bigger  $\alpha$  will result in fuzzier clusters. Typically,  $1 \leq \alpha \leq 2$ .

The minimisation of (8) is accomplished by iteratively updating the cluster centres and FMVs with the following functions:

$$\mathcal{C}_c = \frac{\sum_{m=0}^{M-1} \mu_{m,c}^{\alpha} \mathcal{F}_m}{\sum_{m=0}^{M-1} \mu_{m,c}^{\alpha}} \quad (9)$$

$$\mu_{m,c} = \left( \sum_{\tilde{c}=0}^{C-1} \left( \frac{\delta(\mathcal{F}_m, \mathcal{C}_c)}{\delta(\mathcal{F}_m, \mathcal{C}_{\tilde{c}})} \right)^{2/(\alpha-1)} \right)^{-1} \quad (10)$$

#### 4.2 Distance metrics

Whereas previous works primarily used the standard Euclidean distance metric:

$$\delta_{\text{Euclid}}(\mathcal{F}_m, \mathcal{C}_c) = \|\mathcal{F}_m - \mathcal{C}_c\|_2^2 \quad (11)$$

with  $\|\cdot\|_2$  is the  $\ell_2$  norm of a vector, we investigate, here, the cosine distance as well:

$$\delta_{\text{Cos}}(\mathcal{F}_m, \mathcal{C}_c) = 1 - \frac{\mathcal{F}_m^T \mathcal{C}_c}{\|\mathcal{F}_m\|_2 \|\mathcal{C}_c\|_2}. \quad (12)$$

This choice of similarity metric also derives from work on speaker verification. For speaker embeddings extracted by the ECAPA-TDNN, the closeness of two embedding vectors is related chiefly to their *direction and orientation* because of the AAM loss function used. In a similar manner, since the mod-MFCC features should ideally be scale-invariant, the cosine distance is applicable here as well and, as we demonstrate, turns out to be more discriminative.

#### 5 Cluster-based source separation

The separation framework used here is *identical* to that described in [5, 10]. The main steps are as follows: first, we obtain an initial estimate of the target source in each cluster by means of time-frequency masking (Section 5.1). These initial estimates are then used to time-align the microphone signals in the respective clusters. Following, a simple delay-and-sum beamforming (DSB) is applied to compute the enhanced target signal for the cluster (Section 5.2). Additionally, the fuzzy membership

values will be exploited to perform a weighted delay-and-sum beamformer, termed fuzzy membership value aware DSB (FMVA-DSB) (Section 5.3). As the last step, the improved source estimates are used to compute a post-filter (Section 5.4), which is applied to the beamformed signals for additional noise and interference suppression. These steps are schematically depicted in Fig. 1

While this is a relatively simple framework, it is still insightful because the quality of the speaker separation is directly correlated with the cluster quality. Additionally, it allows a straightforward possibility to include the fuzzy membership values within the framework — which gives more insight into the clustering. This makes the framework a good tool for evaluating the clustering. Note that this does not gainsay the importance of more sophisticated methods, e.g., using cross channel correlations [22, 23] to statistically optimise the separation. Only, this is not fully relevant to the scope of the current study (improving the clustering), and can be tackled in future work. We can reasonably expect a good clustering to improve the performance of the more sophisticated methods as well.

### 5.1 Initial source estimation

The time-frequency (T-F) masks —  $\mathcal{M}(l, k)$  — used for the initial estimate are obtained based on the empirically validated assumption that localised speech sources are approximately W-disjoint in their STFT representation [24]. In order to compute this mask, we assume the

amplitude at T-F bins from microphones close to the target sources is greater than the amplitude of the microphones close to other sources or the background microphones. Thus, if we choose a reference for each source, we can compare their amplitudes to obtain a *rough* indication of which T-F bins are dominated by which source. By including information from the reference microphone allocated to the background cluster additionally helps to suppress reverberation and noise in the initial estimate.

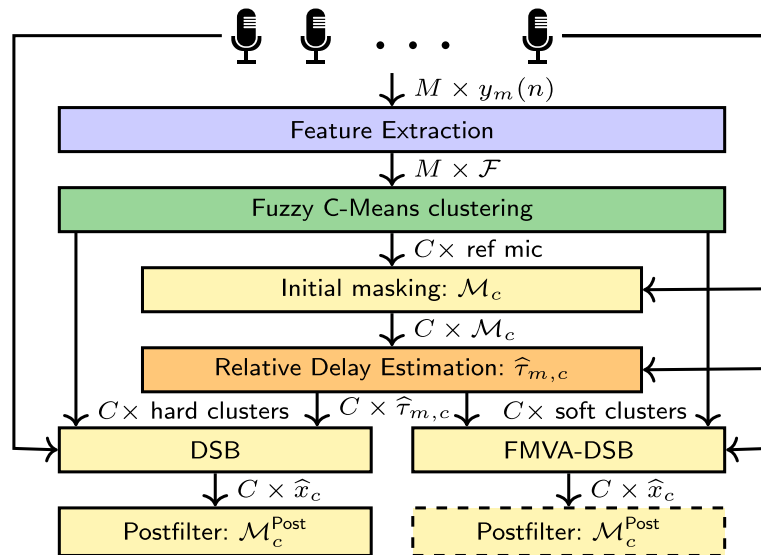
We can directly use the FMVs to select the reference microphone  $Y_c^{\text{ref}}(l, k)$  of each cluster  $c$ . This is simply done by selecting the microphone with the highest fuzzy value for that cluster:

$$Y_c^{\text{ref}}(l, k) = Y_m(l, k) \text{ if } \mu_{m,c} > \mu_{\bar{m},c}, \\ \forall \bar{m} \in \{0, \dots, M-1\}, \bar{m} \neq m \quad (13)$$

Now that we have chosen a reference signal for each cluster, the respective binary mask,  $\mathcal{M}_c(l, k)$ , is obtained by comparing the amplitude of each T-F bin of the reference signals:

$$\mathcal{M}_c(l, k) = \begin{cases} 1 & |Y_c^{\text{ref}}(l, k)| > \frac{1}{B} \sum_{b=l-B+1}^l |Y_{\bar{c}}^{\text{ref}}(b, k)|, \\ & \forall \bar{c} \in \{0, \dots, C-1\}, \bar{c} \neq c \\ 0 & \text{else.} \end{cases} \quad (14)$$

Here, we have introduced the averaging parameter  $B$  which, while not required for conventional binary masking, is necessary for the ASN setting. This is because the



**Fig. 1** Scheme for clustering and cluster based source separation. Features (either Mod-MFCCs or speaker embeddings) extracted from the microphone signals are used to cluster the microphones. Inter- and intra-cluster information is then exploited to extract the sources dominant in each speech cluster. Yellow blocks indicate stages at which speaker separation can be performed — and which we use for evaluation. These consist of initial masking, delay and sum beamforming (DSB), fuzzy membership value aware DSB (FMVA-DSB) and postfiltering one of the DSB outputs. The dotted box is a condition that is not included in the tabulated results

inter-microphone delay for a source is non-negligible compared to the STFT length and frameshift due to the much larger microphone spacings. These delays induce jitter in the STFT amplitudes, and consequently would do the same to the masks without averaging.

The obtained masks of a cluster  $c$  can then be applied to the microphone signals to get the source estimate  $\hat{X}_{m,c}^{\text{Mask}}(l, k)$  of that cluster:

$$\hat{X}_{m,c}^{\text{Mask}}(l, k) = \mathcal{M}_c(l, k) Y_m(l, k) \quad (15)$$

## 5.2 Mask-based delay-and-sum beamforming

The mask can already extract the corresponding source from the mixture at each microphone. However, masks are inherently non-linear operations and combined with the crude definition of the initial mask results in sub-par quality and intelligibility of the masked signals. A better signal estimate can be obtained by a simple delay and sum beamformer. In contrast to compact microphone arrays, the inclusion of more microphones does not necessarily improve the separation capability of the beamformer [4]. Therefore, only microphones with sufficient target dominance should be considered — and this information is reflected in the FMV.

Thus, to attribute microphones to a cluster, we transform the fuzzy clusters into hard partitionings based on the FMV. A microphone  $m$  is allocated to cluster  $c$  if:

$$\mu_{m,c} > \mu_{m,\bar{c}}, \quad \forall \bar{c} \in \{0, \dots, C-1\}, \quad \bar{c} \neq c. \quad (16)$$

We will denote the corresponding signal as  $y_{m,c}$ , and  $M_c$  the number of microphones in cluster  $c$ .

To compensate for the inter-microphone delays, we first have to estimate these. For this, the masks,  $\mathcal{M}_c(l, k)$ , are applied to all the microphone signals of the respective cluster — yielding an initial estimate of the underlying source signal of *that* cluster. The delay  $\hat{\tau}_{m,c}$  with respect to the reference microphone of cluster  $c$  is then computed from these estimates by simple correlation analysis. Time-alignment is then performed on the *unprocessed* microphone signals  $y_{m,c}$ , following which the DSB is computed for cluster  $c$ :

$$\hat{x}_c^{\text{DSB}}(n) = \frac{1}{M_c} \sum_m y_{m,c}(n - \hat{\tau}_{m,c}). \quad (17)$$

Note that the original microphone signals, and not the masked signals, are used in (17) since we do not want the distortions caused by the masks in the beamformer output.

## 5.3 FMV-aware delay-and-sum beamforming

As an extension to the DSB, [5] proposed a fuzzy membership value aware DSB (FMVA-DSB). This better

exploits the information given by the FCM where, ideally, the microphones best capturing a source will have high FMV for that source cluster. Thus, the FMVA-DSB output is obtained by a straightforward modification of (17) to yield the weighted sum:

$$\hat{x}_c^{\text{FMVA-DSB}}(n) = \frac{1}{\sum_m \mu_{m,c}} \sum_m \mu_{m,c} y_{m,c}(n - \hat{\tau}_{m,c}). \quad (18)$$

Note that despite the soft weighting applied in (18), the  $y_{m,c}$  are still only the signals of microphones that are ‘hard-clustered’ to cluster  $c$ .

## 5.4 Postfiltering

Similar to the initial mask, a binary mask can be computed to remove leftover interference and noise. This is particularly useful for the lower frequencies since those are hard to improve with simple beamforming. The post-filter is computed on the output of the DSB (or FMVA-DSB) as follows:

$$\mathcal{M}_c^{\text{Post}}(l, k) = \begin{cases} 1 & |\hat{X}_c^{\text{B}}(l, k)| > \frac{1}{B} \sum_{b=l-B+1}^l |\hat{X}_c^{\text{B}}(b, k)|, \\ & \forall \bar{c} \in \{0, \dots, C-1\}, \bar{c} \neq c \\ 0 & \text{else,} \end{cases} \quad (19)$$

where  $\hat{X}_c^{\text{B}}(l, k)$  is the STFT representation of the beamformed signal at source cluster  $c$ . This postfilter is subsequently applied to the *beamformed* signal in a similar manner to (15).

# 6 Experimental study

## 6.1 Focus of the study

Our prior work demonstrated the benefit of speaker embeddings for microphone clustering using simulated scenarios based on shoe-box acoustic models. This served as a proof-of-concept study, and raised the following interesting questions:

- Q1. What is the clustering performance in realistic room environments (speakers of varying loudness, real room responses,...)?
- Q2. What is the effect on the choice of the distance metric used in the clustering?
- Q3. How does the clustering performance degrade as the sources are in closer proximity?
- Q4. How does the time-scale of data aggregation affect the performance?
- Q5. Given that speaker-embeddings are talker-specific, can this be exploited to detect known talkers and only extract them in realistic, *dialogue-like* situations?

These are addressed through the experimental evaluation.

## 6.2 Realistic setup — SINS database

For the evaluations, we make use of the realistic room impulse responses (RIRs) available in the SINS database [15]. This database is based upon the apartment layout and properties of the apartment used in [25], which is depicted in Fig. 2. As can be seen, there is a big living area (with an open kitchen), a bedroom, a bathroom, a toilet and a hall. The total floor area is  $50m^2$ . CATT-Acoustic with cone-tracing [26] is used to compute the RIRs for different combinations of source and microphone positions. This is an important step towards validation of the system in real world settings, and a step up from shoe-box acoustics used in our previous work. In turn, this might validate the usefulness of shoe-box simulation as an evaluation setup if the results stay consistent!

To better interpret the performance of the system, we split the scenarios into two sets based on the inter-source distance. The first set of scenarios — designed to answer Q1 — is a direct parallel to what we previously did using shoe box acoustics. The scenario only selects sources and microphones from within the living (and kitchen) area, where one source is in the left half of the room, and another one in the right half. For maximum interpretability, we avoid cases where the critical distance regions of the sources can overlap.

The second set of scenarios increases the difficulty of microphone-cluster assignment by bringing the sources

closer to each other. In this setting, sources are separated by at *most* three times the critical distance of the room, while the minimum distance is limited by the dataset to  $0.4m$ . Since the critical distance of the room is  $d_{crit} = 0.68m$ , the critical distance regions of the sources will overlap. The performance in such situations will provide us with an answer to Q3.

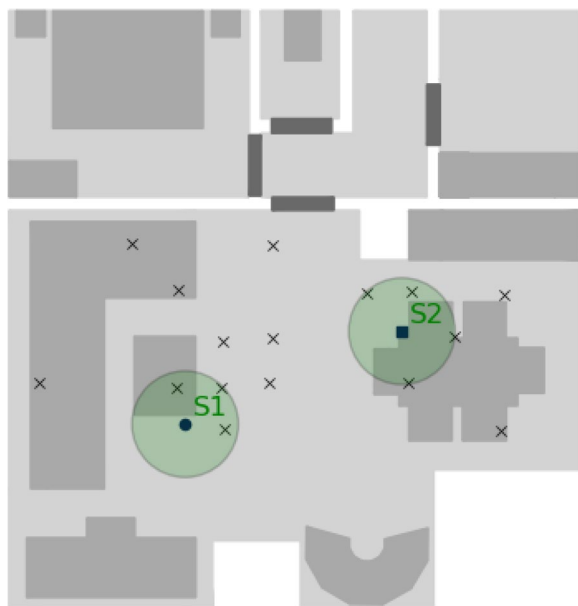
In both scenarios, we shall test the Euclidean metric as well as the cosine distance metric — the point of Q2.

To answer Q4 — how the segment length for feature extraction influences the resulting clusters — we will revert to the first set of scenarios, to reduce the influence of other factors. We will take 4 s as our baseline, consistent with prior work in the literature, and benchmark the performance here against segment lengths of 2, 1 and 0.5 s.

Lastly, for Q5, we incorporate a known speaker embedding into the clustering algorithm. For this, we generate a scenario where the interfering speaker is constantly active, whereas the known speaker is active only for a short time in the middle of the scenario. Since the speaker is *known*, we initialise one cluster centre using the pre-computed speaker embedding of the known speaker. While the target source is inactive, we should ideally have an empty cluster for this source, while the cluster should be populated by microphones during the period of source activity.

For each scenario, there are 200 different settings with  $M = 16$  microphones distributed across the room, and the presence of  $J = 2$  sources for each setting. Furthermore, we ensure that at least 3 microphones are picked from within the critical distance of each source, while the locations of the other  $16 - 3J$  microphones are chosen at random.

The database consists of four-element microphone array nodes. Since we consider individually distributed microphones, we only pick one microphone from each node. We do, however, select a random microphone in order to increase the diversity in the scenarios.



**Fig. 2** SINS room for a specific scenario. The solid dots indicate the location of the two sources, while the crosses are the microphone positions. The green circles indicate the critical distance region for each source ( $d_{crit} = 0.68m$  for the room).

## 6.3 Audio data

The LibriSpeech corpus [27] is chosen for the dry speech sources in the experiments. In line with previous work ([28]): signals of 10 s are selected from the train-clean-100 LibriSpeech subset, where a voice activity detector is used to verify the presence of speech in the selected segments. The corpus contains recordings of different speakers *and at different amplitudes*. We do *not* normalise the utterances to equal levels — thus allowing for combinations of speakers where one speaker can be up to 12 dB louder than the other.



The ECAPA-TDNN is trained on the Voxceleb 1 and 2 database [29], where audio of around 7250 celebrities, and in different environmental settings, is scraped from YouTube. Thus, it is trained on *completely different* data than that used in the evaluation.

#### 6.4 Parameter settings

All audio signals are sampled at 16kHz. A von Hann window of length 512 samples (32ms) and window shift of 160 samples (10ms) is applied before computing the STFT representations. The MFCC parameters are:  $L = 16$  and  $Q = 8$ . Discarding the zeroth MFCC-bin, we take the first  $N = 13$  elements, resulting in a 39-dimensional feature vector  $\mathcal{F}^{MFCC}$ .

The speaker verification feature  $\mathcal{F}^{SpVer}$  length is 192. However, we note that a longer feature vector does not necessarily lead to more informative features for the Mod-MFCC feature representation. The averaging factor  $B$  for the mask computation in (14) and (19) is set to 5. For clustering, we use the fuzzy C-means python package [30].

#### 6.5 Evaluation metrics

Defining good performance metrics that can quantify the clustering quality is not straightforward as it is difficult to define a ground truth.

Attempts have been made to generate ground truths with the help of oracle knowledge of either microphone-source distances [4, 6] or the RIRs [3]. However, the former fails to convey the full picture regarding the signal mixing (it considers strictly circular boundaries without, e.g., accounting for the sound propagation along indirect paths). Using the oracle RIRs for the ground truths does solve the problem of creating non-circular boundaries, but is not easily adaptable to include background clusters or variations in signal levels.

Generating such ground truths also has the disadvantage of forcing hard cut-offs — which does not well-describe the soft transition between clusters. The normalised cluster-centroid-to-source distance metric used in [9] does give more informative results in that sense. However, it also does not convey the full picture of the signal mixture (e.g. if one source speaks louder than the other one), and thus assumes that a circular distribution around the target speaker is the ideal result.

Therefore, we proposed 3 additional metrics in [13], which should provide an intuitive means of quantifying the clustering. This is briefly discussed in Section 6.5.1. We also note that an indirect way to evaluate the cluster quality is by evaluating the performance of the subsequent tasks, e.g. [2] evaluates the performance based on the results of a gender classification task. In this paper,

we evaluate the clusters based on standard instrumental metrics for speaker separation, which will be explained in Section 6.5.2.

##### 6.5.1 Metrics to evaluate clustering quality

The goal of our 3 alternative metrics is to allow an intuitive interpretation of the clustering performance. Since the underlying aim is source separation, a clustering that favours microphones with a strong direct-path component and a good signal to interference and noise ratio would be desirable. Accordingly, we compute (i) the direct-to-reverberant ratio (DRR) and (ii) the direct-to-reverberant, interference, and noise ratio (DRINR) for each microphone  $m$  allocated to a *speech-source* cluster. To this end, we split source signal  $x_{j,m}(n)$  into the direct path component  $x_{j,m}^{\text{dir}}$  and the reflections  $x_{j,m}^{\text{rev}}$ :

$$x_{j,m}(n) = x_{j,m}^{\text{dir}}(n) + x_{j,m}^{\text{rev}}(n). \quad (20)$$

Then the DRR and DRINR are defined as follows:

$$\text{DRR} = \frac{\sum_n (x_{c,m}^{\text{dir}}(n))^2}{\sum_n (x_{c,m}^{\text{rev}}(n))^2} \quad \text{and} \quad (21)$$

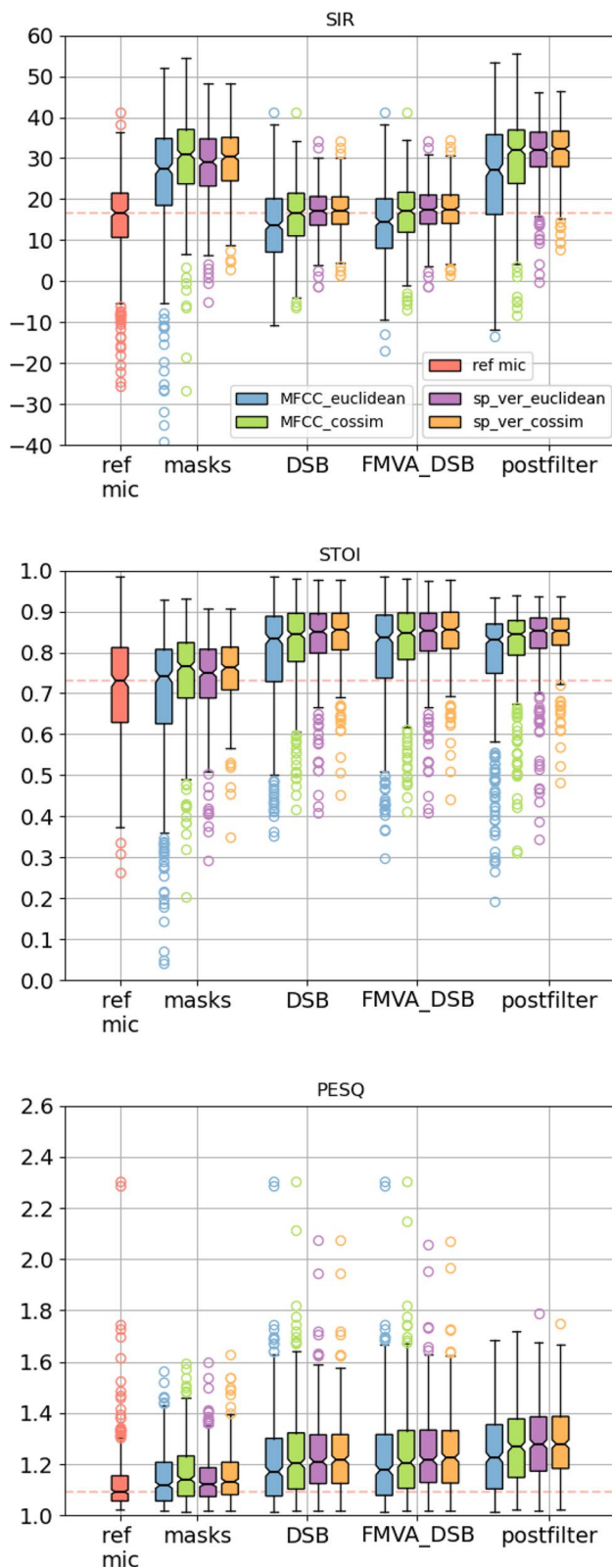
$$\text{DRINR} = \frac{\sum_n (x_{c,m}^{\text{dir}}(n))^2}{\sum_n (y_m(n) - x_{c,m}^{\text{dir}}(n))^2} \quad (22)$$

Subsequently, we plot the *distribution* of these values. A distribution centred around high DRRs and DRINRs values indicates that the clustering selects only those microphones with relevant information about the speaker of that cluster. Lastly, the third metric indicates the amount of spatial diversity available from the clustering. This is computed as the average number of microphones allocated to a speech cluster.

##### 6.5.2 Source separation metrics

As previously noted, clustering quality is indirectly reflected by performance in the subsequent tasks. Here, we use source separation metrics for this purpose, under the reasonable assumption that good clusters would lead to good source separation. We consider 3 standard and widely used instrumental metrics for source separation: the first is the source-to-interference ratio (SIR), as defined by [31]. This is an important metric for the initial masks since the masked signals are used to estimate the TDOA for subsequent delay compensation in the DSBs. After applying the initial masks, it is crucial that only the target source is present for a correct TDOA estimation.

However, interference and noise suppression is only a part of the story. We also use the short-time objective intelligibility (STOI) [32] and the perceptual evaluation



**Fig. 3** Performance metrics (SIR, PESQ, STOI) showing the separation effectiveness of the cluster feature types (colours) and method (x-axis) for the first set of scenarios, where the sources are always sufficiently far apart. For this and the other scenarios, some audio examples are available at [https://users.ugent.be/sbkindt/EURASIP\\_ASN/](https://users.ugent.be/sbkindt/EURASIP_ASN/)

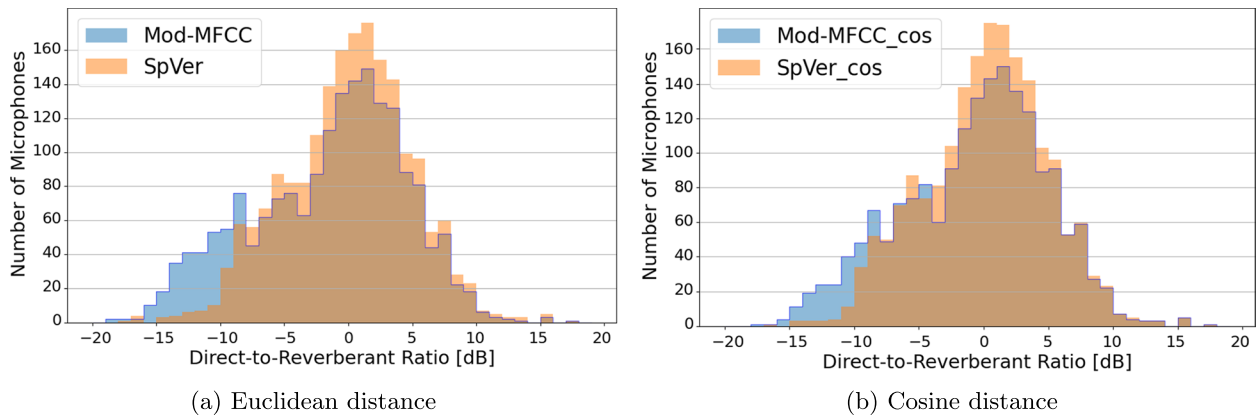
of speech quality (PESQ) [33]) metrics to quantify the target-attenuation.

## 7 Results and discussion

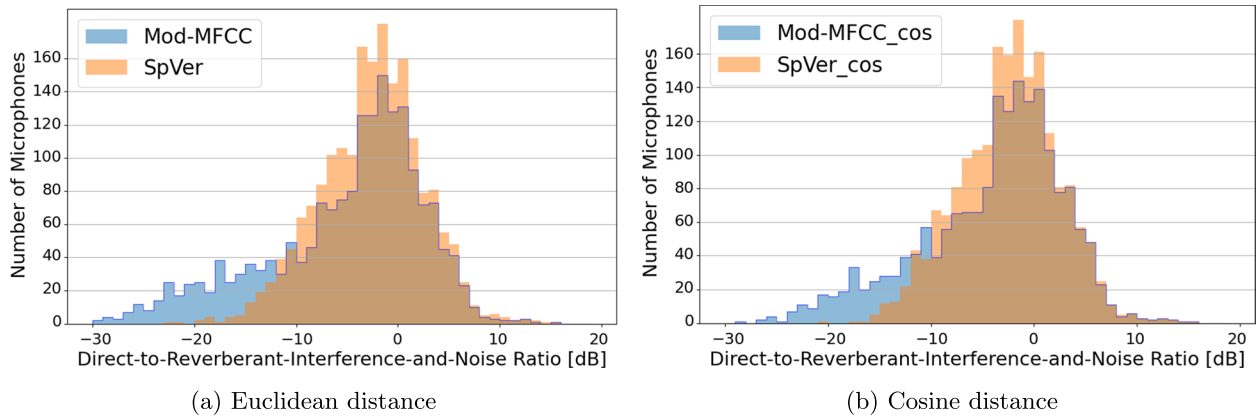
### 7.1 First set of scenarios — sources far apart

The results are plotted in Figs. 3, 4, 5 and Table 1. We first take a look at the results for the Euclidean distance only, since that corresponds with the results for the shoe-box acoustics presented in [13]. Here, we see fairly similar patterns: in Fig. 3, the notched box-plots show that the speaker verification features lead to better speaker separation metrics, with statistical significance at the median level. This is true for all separation methods (x-axis) and evaluation metrics (subfigures). Figures 4a and 5a also show that the Mod-MFCC features tend to include more microphones with relatively low source dominance (low DRR and DRINR). In contrast, the histogram plots for the speaker embeddings are narrower and include a larger number of microphones with relatively high DRRs and DRINRs, suggesting that using the speaker embeddings allows the clustering to find more useful microphones. Additionally, Table 1 also indicates that the cluster size when using speaker embeddings is larger than that using mod-MFCC features. This combination of a larger number of microphones which have, on average, better source dominance (high DRR and DRINR), indeed makes it possible to improve separation — which is seen in the separation metrics. For comparison, the metrics computed on the reference microphone for each cluster is also plotted (first column).

Interestingly, when using the cosine distance metric, the performance of the Mod-MFCC features improves greatly and the separation performance becomes comparable to the separation performance when using speaker embedding based features. The improvement is less evident for the speaker-embedding based features. The DRR and DRINR distributions in Figs. 4b and 5b indicate, the speaker embeddings in combination with the cosine distance yield ever so slightly narrower histograms compared to using the Euclidean distance. This may be verified more straightforwardly from the DRINR histograms for Euclidean and cosine distance in Fig. 6. Thus, the tendency is towards the selection of fewer lower-quality microphones for each source cluster. This improved microphone selection translates to, similarly, a slightly



**Fig. 4** DRR histograms with **a** the Euclidean distance or **b** the cosine distance for the first set of scenarios. In this set, the sources are located quite far apart. The DRRs are computed only for microphones that are part of a source cluster



**Fig. 5** DRINR histograms with **a** the Euclidean distance or **b** the cosine distance for the first set of scenarios. In this set, the sources are located quite far apart. The DRINRs are computed only for microphones that are part of a source cluster

**Table 1** Average number of microphones per *source* cluster and choice of distance metric. Results for the first set of scenarios, where the sources are placed relatively far from each other. The larger the number of microphones, the more the spatial diversity available for a cluster

	Euclidean	Cosine
MFCC	4.64	4.57
SpVer	4.84	4.76

better separation performance, visible in Fig. 3, where there are fewer outliers and a more compact boxplot.

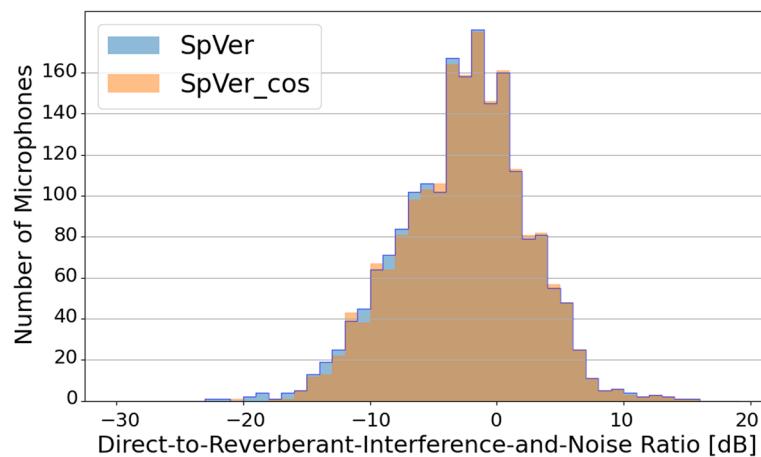
When comparing the average number of microphones per source cluster (Table 1), there are fewer microphones on average when using the cosine distance. However, the DRR and DRINR distributions indicate that the microphones that are omitted are mostly of lower quality.

In general, we can conclude that for this set of scenarios, the cosine distance metric is better than the

Euclidean distance. The improvement is most marked for Mod-MFCC based features. The combination of cosine distance and Mod-MFCC based features yields clusters with *separation* performance comparable to that using speaker embedding based features. Also, we obtain the same trends in performance in this realistic setting as we obtained using the simulated (shoe-box acoustics) rooms.

## 7.2 Second set of scenarios — sources in close proximity

The results for this more challenging setting are presented in Figs. 7, 8, and 9 and Table 2. Bringing the sources closer, unsurprisingly, makes the clustering harder. While the separation performance using speaker embedding features still outperforms the mod-MFCC-based features, all speaker separation metrics are lower than for the first scenario. SIRs are even, sometimes, below 0 dB. However, since the sources can have different signal amplitudes, it is possible that for such close sources, one source dominates, making it nearly impossible to separate those with the chosen simple separation



**Fig. 6** DRINR histograms of the first set of scenarios (sources are far) for the speaker embedding based clustering. The DRINRs are computed only for microphones that are part of a source cluster

scheme. This highlights the importance of more sophisticated separation approaches.

One interesting observation on the speaker separation metrics is that the cosine distance seems to give a significant improvement over the Euclidean distance, and for *both* sets of features. This is most prominent for the initial mask estimate, which indicates that using the cosine distance yields a better *reference microphone* for each cluster.

The histograms in Figs. 8 and 9 tell a similar story, where the speaker embedding features select more useful microphones. Nevertheless, it is instructive to zoom in on the region of less than ideal microphones ( $-10$  dB DRR and lower) in Fig. 8a. There, the DRR histogram would suggest that the Mod-MFCC features lead to a better microphone allocation than the speaker embeddings. However, when looking at the DRINR distribution in Fig. 9a, the conclusions seem to be reversed. This indicates that the speaker embeddings are better at incorporating information about the target and interference speaker for the clustering, rather than only the distance of a microphone to the target speaker (which is likely what the Mod-MFCC based features focus on). Note that this is mainly for the Euclidean distance metric. The results are more consistent when using cosine similarity. Additionally, Fig. 10 does demonstrate a clear benefit of the cosine distance in combination with speaker embeddings, making the cosine distance more beneficial in situations where the sources are close compared to situations where the sources are distributed further apart in the room (Fig. 6).

Table 2 shows that for the Euclidean distance, a similar conclusion as for the previous sections is applicable: speaker verification features generate slightly larger

clusters, and of higher quality (seen from the DRR and DRINR histograms). For the cosine distance, the number of microphones does not significantly change between the choice of features. Again, the number of microphones decreases when using the cosine distance, but it is mainly the lower quality microphones that are removed (conclusion from the DRINR plots in Fig. 9).

### 7.3 Effect of segment length

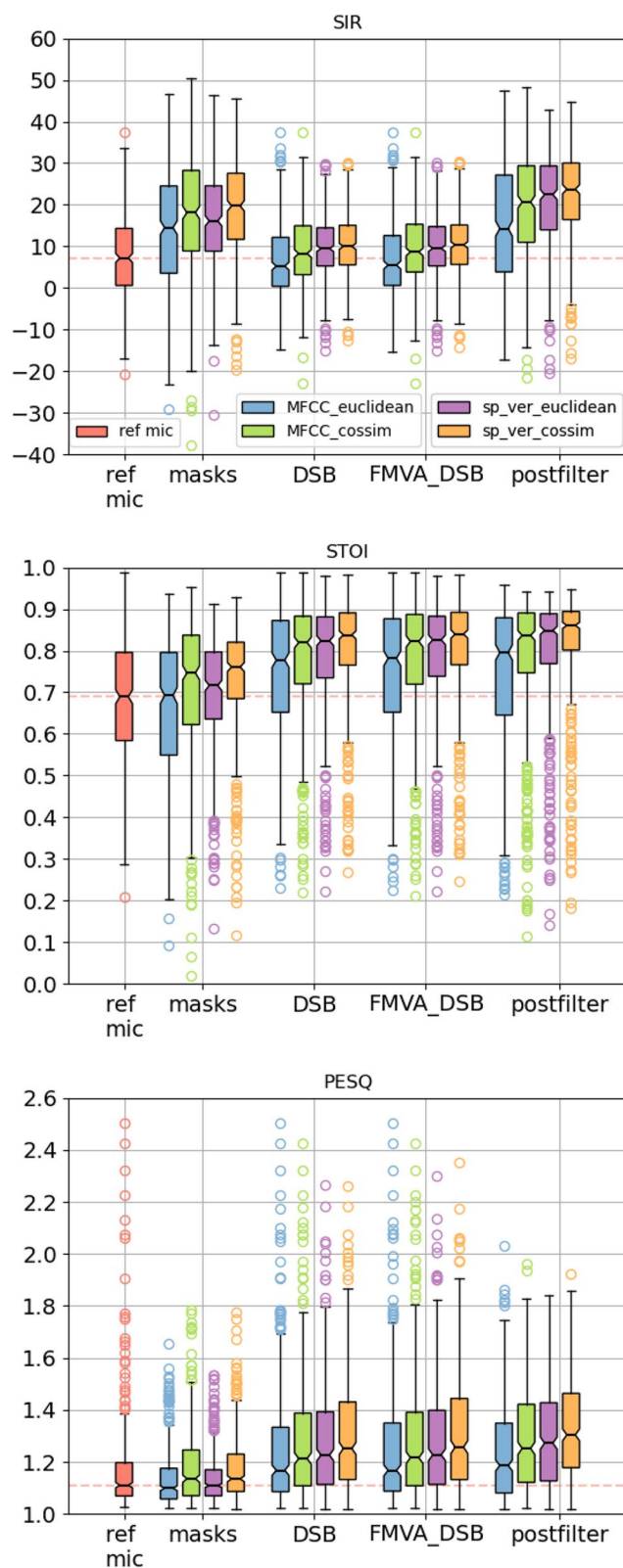
Figures 11, 12, and 13 and Table 3 show the impact of shortening the length of the segment of the signal given to the feature extractors. The experiments were carried out for the same set of scenarios as Section 7.1 and using only the cosine similarity, since it yielded the best results in the previous experiments.

For the Mod-MFCC features, the clusters consistently degrade as the segment lengths decrease and more drastically for lengths of 1 s and 0.5 s. In contrast, for the speaker embedding features, the segment length seems to have only a marginal impact on the clustering capability, even for the short length of 0.5 s. This is further visible in both the DRR and DRINR distributions, where those for the speaker embeddings have only a very slight shift towards lower DRRs and DRINRs, while for the MFCC-based features, the shift is marked, becoming increasingly prominent for shorter evaluation lengths.

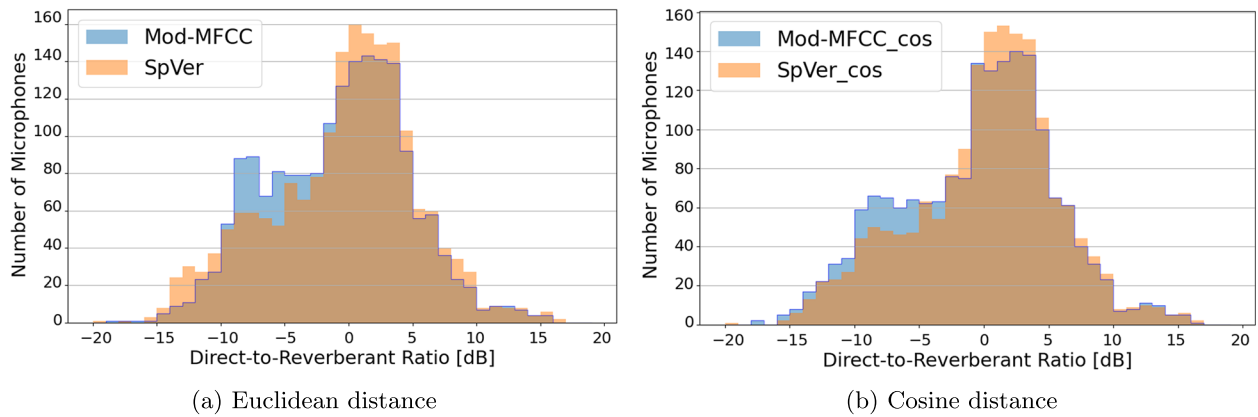
The same effect is visible in the speaker separation metrics in Fig. 11: the performance of the Mod-MFCC based features again starts dropping with lower segment lengths. In contrast, the performance of the embedding based speaker separation stays quite consistent.

In terms of the average number of microphones per cluster — this does not drastically change for different

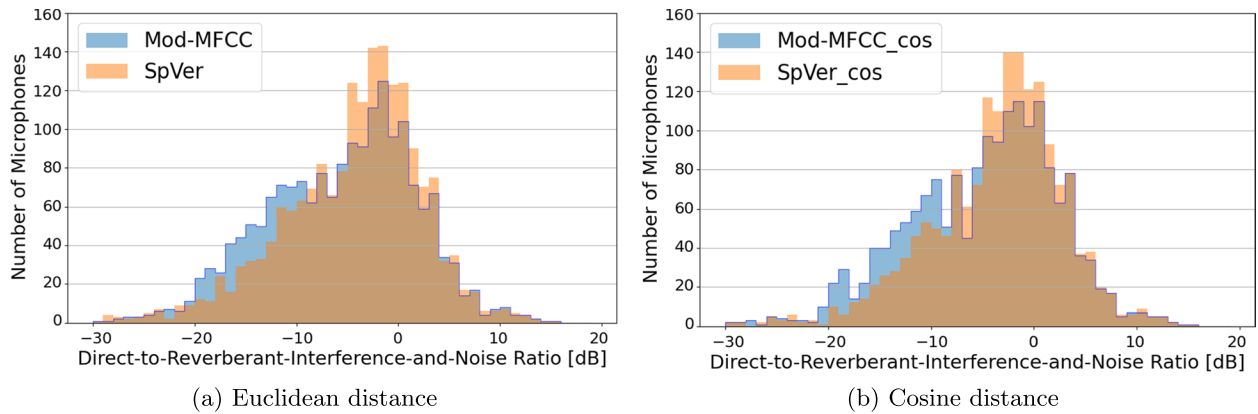




**Fig. 7** Performance metrics (SIR, PESQ, STOI) showing the separation effectiveness of the cluster feature types (colours) and method (x-axis) for the second set of scenarios, where the sources are maximally separated by three times the critical distance



**Fig. 8** Histograms of DRR with **a** the Euclidean distance or **b** the cosine distance for the second set of scenarios. In this set, the sources are separated by at most three time the critical distance. The DRRs are computed only for microphones that are part of a source cluster



**Fig. 9** Histograms of the DRINR with **a** the Euclidean distance or **b** the cosine distance for the second set of scenarios. In this set, the sources are separated by at most three time the critical distance. The DRINRs are computed only for microphones that are part of a source cluster

**Table 2** Average number of microphones per *source* cluster and choice distance metric for the second set of scenarios, where the sources are placed in close proximity to each other

	Euclidean	Cosine
MFCC	4.54	4.39
SpVer	4.64	4.34

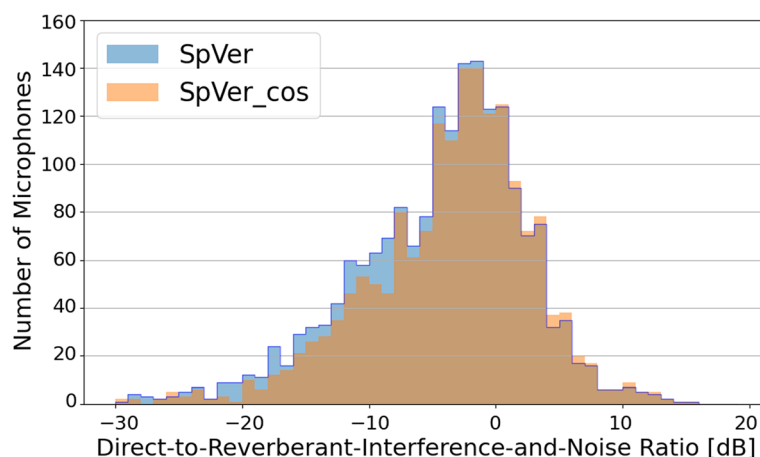
evaluation lengths for the speaker embeddings, while for the Mod-MFCC, the number of microphones *increase*. This larger cluster mainly contains microphones with a poor signal to interference-and-noise ratio (visible in the DRINR histogram Fig. 13a), which negatively impacts the separation performance.

This leads us to the satisfactory conclusion that speaker embeddings computed on short segments still yield robust features for clustering ad hoc distributed

microphones. This can be utilised to lower the computational complexity (less frequent updates of the cluster and feature extraction on short segments). Alternatively, the robustness of the features to shorter segment lengths can be exploited to quickly adapt the clustering in more dynamic scenarios.

#### 7.4 Known speaker embedding

Having shown the general robustness of the speaker embedding features, applied to the task of clustering ad hoc distributed microphones, we focus on investigating whether these features can be exploited to focus on only a *desired* subset of speakers, which are known a priori. We only show one example for this scenario — more as an empirical proof-of-concept. Since the scenario is dynamic, we use shorter evaluation segments of length 2 s. Figure 14 shows the results for this scenario.



**Fig. 10** DRINR histograms for the second set of scenarios (sources are close) for the speaker embedding based clustering. The DRINRs are computed only for microphones that are part of a source cluster

In the first and last part (Fig. 14a and c), where only the interferer is active, there is an empty source cluster for the target source. The FCM generates two other cluster centres that model the interfering source and the background characteristics. Note that in these periods it is desirable that the microphones close to the inactive speaker are grouped in the background cluster. During the period when the target source is active, the features extracted from microphones close to the source match the known-speaker embedding (which is used to initialise the cluster centre for this source) and the FCM faithfully attributes the appropriate microphones to this source — as can be seen in Fig. 14b. This demonstrates, in addition to robustness, the ability to induce microphone clustering in a speaker selective manner. This constitutes an additional benefit of using embeddings as clustering features. Such behaviour would not be straightforward to implement using other features, e.g. those purely based on room characteristics.

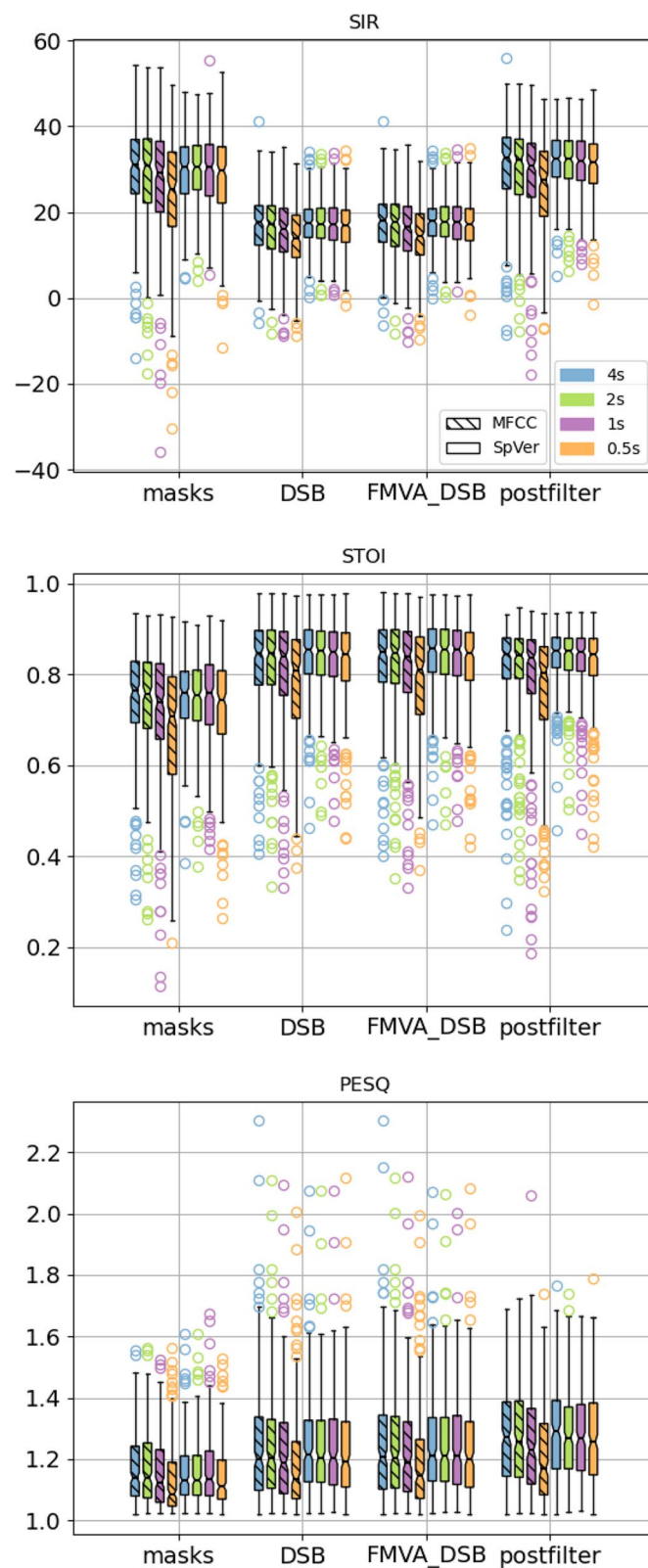
## 8 Conclusions

Our prior work, which introduced speaker embeddings as robust features for clustering ad hoc distributed microphones, raised several interesting follow-up questions that were addressed in this paper. Firstly, we evaluated the performance of speaker embedding features in realistic settings and demonstrated similar trends as previously reported using simulations based on shoe-box acoustics models. Next, the effect of the distance metric used in the clustering algorithm was investigated and it was shown that the cosine distance offers more discriminative clustering compared to the Euclidean metric used previously.

The benefit of this metric was more marked for the baseline mod-MFCC features, bringing their performance to a level comparable to that of the speaker embedding features, in scenarios where the sources are far apart. In more challenging conditions, however, the speaker embedding features, in combination with the cosine distance metric, better exploit the source-specific information and significantly outperform the Mod-MFCC based features.

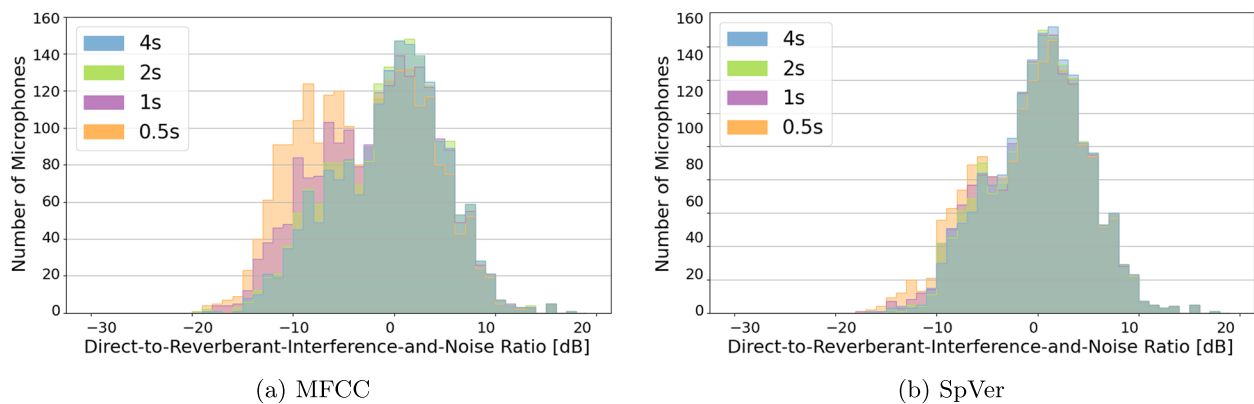
In view of practical implementations, the effect of shorter segment lengths on the clustering performance was studied. Here, whereas mod-MFCC-based features consistently degrade with shorter segment lengths, speaker embedding features are only marginally affected and their performance remains more-or-less constant. Even with a segment length of 0.5 s, the clusters stay similar to the baseline. This robustness of the speaker embedding features can be exploited for two purposes: complexity reduction and/or quicker adaptation in dynamic scenarios. Complexity can be scaled by only computing the features and updating the clusters sporadically and using only a small amount of data, sampled over a wider time-range. To allow for quick adaptation in more dynamic scenarios, the idea would be to similarly compute the features over short, but contiguous time-intervals and update the clusters more frequently.

Lastly, we presented a proof-of-concept of how speaker embeddings could be used to explicitly incorporate information on a known speaker for targeted clustering and separation. In future work, we aim to further focus on this setting and incorporate not only

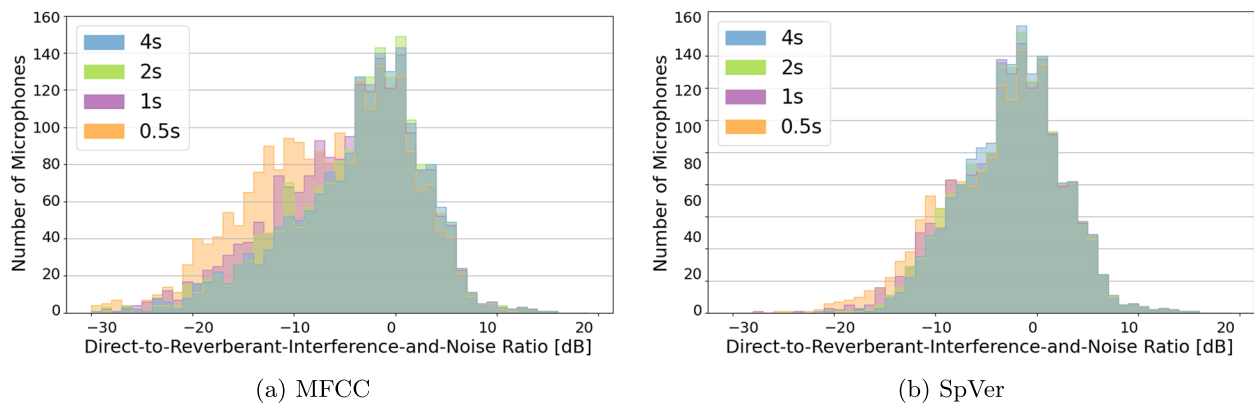


**Fig. 11** Performance metrics (SIR, PESQ, STOI) showing the separation effectiveness of the cluster feature types (hatches), method (x-axis) and duration (colour) for the first set of scenarios, where the sources are always sufficiently far apart





**Fig. 12** Histograms of the DRRs of the **a** Mod-MFCC features and **b** speaker verification features for different evaluation durations and the cosine distance metric. These are computed only for microphones that are part of a source cluster



**Fig. 13** Histograms of the DRINRs of the **a** Mod-MFCC features and **b** speaker verification features for different evaluation durations and the cosine distance metric. These are computed only for microphones that are part of a source cluster

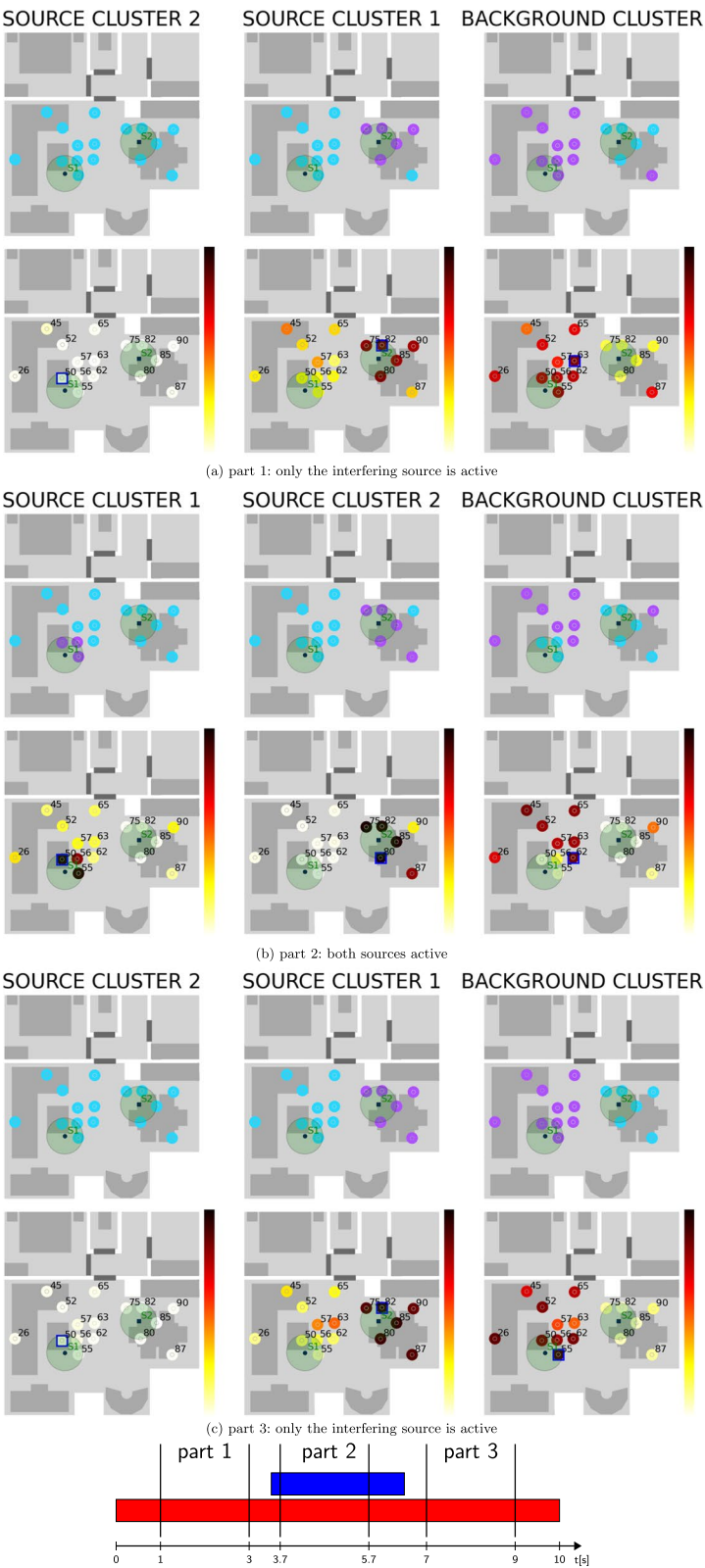
**Table 3** Average number of microphones per *source* cluster for different evaluation lengths. The cosine distance metric is used throughout

	4s	2s	1s	0.5s
MFCC	4.52	4.65	5.50	5.52
SpVer	4.71	4.68	4.77	4.90

more sophisticated separation approaches, like MVDR beamforming [22, 23, 34] or deep learning based methods [35], but also, the improved embedding extractor proposed in [36], which offers increased robustness of the extracted embedding in the presence of interfering speech. Additionally, a comparison with spatially based cluster algorithms, like [3], should be performed, to see the trade-offs between the methods. Investigating

(See figure on next page.)

**Fig. 14** Incorporation of known speaker embedding for targeted clustering. In this scenario, the interfering source is active throughout the experiment. However, the target (known) source is only active for a short period in the middle of the segment. This is schematically indicated above, where the blue bar indicates the time-period where the known speaker is active, while red indicates the interferer activity. We initialise the first cluster with the known speaker embedding. Thereby, the FCM algorithm generates an empty cluster in parts **a** and **c** and only allocates microphones to the target cluster when the known source is active. In the figure, a microphone is part of the cluster if its colour is dark purple, while the light blue colour indicated that it is not part of the cluster



**Fig. 14** (See legend on previous page.)

an optimal combination of spatial and speaker-specific information is also an interesting path we shall explore in future work.

Extra clustering examples and the associated audio corresponding to the presented work are available at [https://users.ugent.be/~sbkindt/EURASIP\\_ASN/](https://users.ugent.be/~sbkindt/EURASIP_ASN/).

#### Abbreviations

AAM	Additive Angular Margin
AMA	Averaged modulation amplitude
ASN	Acoustic sensor network
BLE	Bluetooth Low Energy
CMR	Cepstral modulation ratios
DRINR	Direct-to-reverberant interference and noise ratio
DRR	Direct-to-reverberant ratio
DSB	Delay-and-sum beamformer
ECAPA-TDNN	Enhanced Propagation and Aggregation Time Delay Neural Network
FCM	Fuzzy C-means
FMV	Fuzzy membership value
FMVA	Fuzzy membership value aware
GBF	Geometry-based feature
MFCC	Mel frequency cepstral coefficient
Mod-MFCC	Modulated Mel frequency cepstral coefficient
MSC	Magnitude squared coherence
RIR	Room impulse response
IoT	Internet of things
PESQ	Perceptual evaluation of speech quality
SBF	Signal-based feature
SDLF	Source-dependant latent feature
SE	Squeeze-Excitation
SIR	Signal-to-interference ratio
SRO	Sample rate offset
STO	Sample time offset
STOI	Short-term objective intelligibility
STFT	Short-time Fourier transfer
T-F	Time-frequency
VAD	Voice activity detection
VAE	Variational auto-encoder
WASN	Wireless acoustic sensor network

#### Acknowledgements

The authors thank Rainer Martin (Ruhr-Universität Bochum) for useful discussions on the paper and for providing the realistic RIRs used in the evaluation. Thanks also to Alexander Bohlender from Ghent University (UGent) for the feedback on the paper draft.

#### Authors' contributions

S.K. implemented the methods, carried out the experiments and drafted the paper. J.T. originally designed the ECAPA-TDNN network, gave insight into its working, and added descriptions to the paper. L.B. provided input on the federated learning approach as well as contributed to the SINS simulated RIRs and utilisation code. N.M. conceptualised the idea to use speaker embeddings and gave comprehensive feedback on the writing.

#### Funding

This work is supported by the Research Foundation - Flanders (FWO) under grant number G081420N and imec.ICON: BLE2AV (support from VLAIO). Partners: Imec, Televic, Cochlear, and Qorvo.

#### Availability of data and materials

The evaluation scenarios used and/or analysed during the current study are available from the corresponding author upon reasonable request. The RIR dataset should be requested from the owners of the SINS dataset.

#### Declarations

##### Ethics approval and consent to participate

Not applicable.

##### Consent for publication

Not applicable.

##### Competing interests

The authors declare that they have no competing interests.

Received: 25 March 2023 Accepted: 6 October 2023

Published online: 31 October 2023

#### References

1. A. Bertrand, in *2011 18th IEEE symposium on communications and vehicular technology in the Benelux (SCVT)*, Applications and trends in wireless acoustic sensor networks: A signal processing perspective (IEEE, 2011), pp. 1–6
2. S. Gergen, A. Nagathil, R. Martin, Classification of reverberant audio signals using clustered ad hoc distributed microphones. *Sig. Process.* **107**, 21–32 (2015)
3. A.J. Muñoz-Montoro, P. Vera-Candeas, M.G. Christensen, in *2021 29th European Signal Processing Conference (EUSIPCO)*, A coherence-based clustering method for multichannel speech enhancement in wireless acoustic sensor networks (IEEE, 2021), pp. 1130–1134
4. I. Himawan, I. McCowan, S. Sridharan, Clustered blind beamforming from ad-hoc microphone arrays. *IEEE Trans. Audio Speech Lang. Process.* **19**(4), 661–676 (2010)
5. S. Gergen, R. Martin, N. Madhu, in *Speech Communication; 13th ITG-Symposium*, Source separation by fuzzy-membership value aware beamforming and masking in ad hoc arrays (VDE, 2018), pp. 1–5
6. S. Pasha, Y.X. Zou, C. Ritz, in *2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, Forming ad-hoc microphone arrays through clustering of acoustic room impulse responses (IEEE, 2015), pp. 84–88
7. Y. Zhao, J.K. Nielsen, J. Chen, M.G. Christensen, Model-based distributed node clustering and multi-speaker speech presence probability estimation in wireless acoustic sensor networks. *J. Acoust. Soc. Am.* **147**(6), 4189–4201 (2020)
8. M. Dziubany, R. Machhamer, H. Laux, A. Schmeink, K.U. Gollmer, G. Burger, G. Dartmann, in *2018 26th European Signal Processing Conference (EUSIPCO)*, Machine learning based indoor localization using a representative k-nearest-neighbor classifier on a low-cost iot-hardware (2018), pp. 2050–2054. <https://doi.org/10.23919/EUSIPCO.2018.8553155>
9. S. Gergen, R. Martin, in *Speech Communication; 12. ITG Symposium*, Estimating source dominated microphone clusters in ad-hoc microphone arrays by fuzzy clustering in the feature space (VDE, 2016), pp. 1–5
10. S. Gergen, R. Martin, N. Madhu, in *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, Source separation by feature-based clustering of microphones in ad hoc arrays (IEEE, 2018), pp. 530–534
11. A. Nelus, R. Glitza, R. Martin, in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Estimation of microphone clusters in acoustic sensor networks using unsupervised federated learning (IEEE, 2021), pp. 761–765
12. L. Becker, A. Nelus, R. Glitza, R. Martin, in *2022 International Workshop on Acoustic Signal Enhancement (IWAENC)*, Accelerated unsupervised clustering in acoustic sensor networks using federated learning and a variational autoencoder (IEEE, 2022), pp. 1–5
13. S. Kindt, J. Thienpondt, N. Madhu, in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Exploiting speaker embeddings for improved microphone clustering and speech separation in ad-hoc microphone arrays (IEEE, 2023), pp. 1–5

14. B. Desplanques, J. Thienpondt, K. Demuynck, in *Interspeech 2020*, ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification (International Speech Communication Association (ISCA), 2020), pp. 3830–3834
15. R. Glitz, L. Becker, A. Nelus, R. Martin, In: 2023 31st European Signal Processing Conference (EUSIPCO), Database of simulated room impulse responses for acoustic sensor networks deployed in complex multi-source acoustic environments. (IEEE, 2023), pp. 246–250
16. F. Bimbot, J.F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-García, D. Petrovska-Delacrétaz, D.A. Reynolds, A tutorial on text-independent speaker verification. *EURASIP J. Adv. Signal Proc.* **2004**(4), 1–22 (2004)
17. P.N. Garner, Cepstral normalisation and the signal to noise ratio spectrum in automatic speech recognition. *Speech Commun.* **53**(8), 991–1001 (2011)
18. D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, S. Khudanpur, in 2018 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, X-vectors: Robust dnn embeddings for speaker recognition (2018), pp. 5329–5333. <https://doi.org/10.1109/ICASSP.2018.8461375>
19. J. Hu, L. Shen, G. Sun, in 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Squeeze-and-excitation networks (2018), pp. 7132–7141. <https://doi.org/10.1109/CVPR.2018.00745>
20. J. Deng, J. Guo, N. Xue, S. Zafeiriou, in 2019 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Arcface: Additive angular margin loss for deep face recognition (2019), pp. 4685–4694. <https://doi.org/10.1109/CVPR.2019.00482>
21. J.C. Bezdek, R. Ehrlich, W. Full, FCM: The fuzzy c-means clustering algorithm. *Comput. Geosci.* **10**(2–3), 191–203 (1984)
22. S. Markovich-Golan, A. Bertrand, M. Moonen, S. Gannot, Optimal distributed minimum-variance beamforming approaches for speech enhancement in wireless acoustic sensor networks. *Signal Process.* **107**, 4–20 (2015)
23. D. Cherkassky, S. Markovich-Golan, S. Gannot, in 2015 *23rd European Signal Processing Conference (EUSIPCO)*, Performance analysis of mvdr beamformer in wasn with sampling rate offsets and blind synchronization (IEEE, 2015), pp. 245–249
24. S. Rickard, O. Yilmaz, in 2002 *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, On the approximate W-disjoint orthogonality of speech (IEEE, 2002), p. 529
25. G. Dekkers, S. Lauwereins, B. Thoen, M.W. Adhana, H. Brouckxon, B. Van den Bergh, T. Van Waterschoot, B. Vanrumste, M. Verhelst, P. Karsmakers, in 2017 *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE)*, The sins database for detection of daily activities in a home environment using an acoustic sensor network (IEEE, 2017), pp. 32–36
26. B.I. Dalenbäck. TUCT v2.0e:1, CATT (1999). <http://www.catt.se>. Accessed 2019
27. V. Panayotov, G. Chen, D. Povey, S. Khudanpur, in 2015 *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, Librispeech: an asr corpus based on public domain audio books (IEEE, 2015), pp. 5206–5210
28. A. Nelus, R. Glitz, R. Martin, in 2021 *29th European Signal Processing Conference (EUSIPCO)*, Unsupervised clustered federated learning in complex multi-source acoustic environments (IEEE, 2021), pp. 1115–1119
29. J.S. Chung, A. Nagrani, A. Zisserman, in *Interspeech 2018*, Voxceleb2: Deep speaker recognition (International Speech Communication Association (ISCA), 2018), pp. 1086–1090
30. M.L.D. Dias. Fuzzy c-means: An implementation of fuzzy c-means clustering algorithm (2019). <https://doi.org/10.5281/zenodo.3066222>. <https://git.io/fuzzy-c-means>
31. E. Vincent, R. Gribonval, C. Févotte, Performance measurement in blind audio source separation. *IEEE Trans. Audio Speech Lang. Process.* **14**(4), 1462–1469 (2006)
32. C.H. Taal, R.C. Hendriks, R. Heusdens, J. Jensen, in 2010 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, A short-time objective intelligibility measure for time-frequency weighted noisy speech (IEEE, 2010), pp. 4214–4217
33. A.W. Rix, J.G. Beerends, M.P. Hollier, A.P. Hekstra, in 2001 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2, Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs (IEEE, 2001), pp. 749–752
34. E.A. Habets, J. Benesty, S. Gannot, I. Cohen, in *Speech processing in modern communication*, The MVDR beamformer for speech enhancement (Springer, 2010), pp. 225–254
35. Z.Q. Wang, J. Le Roux, J.R. Hershey, in 2018 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Multi-channel deep clustering: Discriminative spectral and spatial embeddings for speaker-independent speech separation (IEEE, 2018), pp. 1–5
36. J. Thienpondt, N. Madhu, K. Demuynck, in 2023 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Margin-mixup: A method for robust speaker verification in multi-speaker audio (IEEE, 2023)

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)