RESEARCH

# Ad-hoc microphone clustering with speaker embeddings under realistic and challenging scenarios.

Stijn Kindt[1*], Jenthe Thienpondt[1], Luca Becker[2] and Nilesh Madhu[1]

**Abstract**

Speaker embeddings, taken 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 Mod-MFCC based features, using speaker embeddings yielded a more robust clustering of the microphones with benefits for subsequent enhancement stages. This was demonstrated on simulated data based on shoe-box acoustics models. In this work, we present a more thorough analysis of the use of speaker embeddings for microphone clustering, in more realistic settings. Further, 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 increasingly more challenging situations 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. The work here aims to establish speaker embeddings as a robust feature for ad-hoc microphone clustering and present some rule-of-thumb considerations to be taken into account when designing practical systems based on this approach.

**Keywords:** wireless acoustic sensor networks; distributed microphone clustering; clustering metric; ECAPA-TDNN; speaker embeddings; speaker separation

## 1 Introduction

Many 'smart' devices carry at least one microphone. Common 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), gives great coverage over the room. ASNs can perform valuable tasks, like sound source localisation and separation, applicable in a wide range of domains such as surveillance for assisted living and healthcare, hearing aids, communications, etc. [1].

Since these microphones can be distributed all over the room, the spatial diversity is greater than that of a compact microphone array, where multiple microphones are put in close proximity to each other. However, it is not straightforward to combine the distributed microphone signals. The microphones are not driven by the same clock signal: both clock sample rate offsets (SROs) and sample time offsets (STOs) are 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, named wireless ASN (WASN), there are typically also bandwidth and processing power limitations on the system. Furthermore, for transportable microphone-carrying devices, the position of the microphones is not known a priori. Forming a WASN from these 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 how similar their surrounding sounds are. This will then group microphones that are close to the same point of interest, or reversely, group microphones that

*Correspondence: stijn.kindt@ugent.be
[1]IDLab, Department of Electronics and Information Systems, Ghent University - imec, Ghent, Belgium
Full list of author information is available at the end of the article

characterise the interfering or noise sources. These clusters have already been proven valuable for subsequent steps like source classification [2, 3] or source separation [4, 5].

The WASNs cluster procedures consist of two major parts: a clustering algorithm is chosen, and features are designed upon which the algorithm performs clustering. A variety of cluster features have been proposed. 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. [6] first estimate the room impulse responses (RIRs) to then cluster the microphones. Both of these techniques solely use the room properties to perform clustering.

In [3] the MSC on the speech-active parts of the signal is utilised as cluster features. This contains information about the RIRs and the content of the signal, thus both the room characteristics and signal correlations are being used. In [7], similarly, they compute the individual microphone auto-correlation of the source signal and the auto-correlation of the noise signal. The signal is separated into the noise and target components with the help of voice activity detection (VAD).

All mentioned techniques are influenced by the room characteristics. These can be useful if you additionally want to know the position of the microphones in the room. However, for that, you would have to first gather enough samples of positions in the room first, as they did in [8]. In contrast, features that focus on speech-specific content are useful to focus on pre-determined targets (e.g. in smart care homes where you want to monitor particular patients). Also, speaker-specific features can make it easier to cluster all background microphones together, even though they are in different parts of the room. Similarly, they could make it easier to increase or decrease the extent of the cluster depending on the noise level of the room.

Pure signal-dependant features were proposed by [9]. They use a variational auto-encoder (VAE) trained on all types of speech and music data. Then, they randomise the parameters of the bottleneck layer and distribute that model over all the microphone nodes. The network on each node updates the bottleneck weights based on the captured signal, which will essentially overfit on that signal. The updated parameters are sent back to the central node and are used as cluster features. Since this is a federated learning based approach, it preserves the privacy of the speaker. However, the privacy constraint inevitably constrains the use case where you look for a specific speaker. Also, retraining the network at each node comes at a relatively high computational cost, which they have discussed and improved in [10].

Gergen *et al*. [2] proposed features which are based on the Modulation Mel frequency cepstral coefficients (Mod-MFCCs). MFCC mean subtraction is done in order to reduce the effect of the room under the assumption that the source and microphone stay sufficiently static during the evaluation period.

Since deep neural networks have shown great performance increases, we proposed speaker embedding features in [11]. These speaker embeddings are generated by a pre-trained speaker verification network: the Enhanced Propagation and Aggregation Time Delay Neural Network (ECAPA-TDNN) [12]. This network should, for the speaker verification task itself, be robust against reverberation, and hence give room-independent embeddings which serve well as signal-dependant cluster features.

We performed an initial comparison of these embeddings with the Mod-MFCC based features in simulated shoe-box rooms, and showed the more robust and visually logical clusters generated by the embeddings. For this contribution, we want to expand upon the evaluation with more realistic room environments, based on the SINS database [13], as was done in [14]. Expansion of the evaluation will also be done by considering various scenarios: we will bring the sources closer to each other, and shorten 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 can increase performance by either needing less computational power or being able to update the clusters more frequently in case of moving sources or microphones.

For the other major part of clustering, the clustering algorithm, there are many approaches in the literature: K-means clustering is used in [6], non-negative matrix factorisation is utilised in [3], while matrix bi-partitioning is deployed in [9]. In contrast, fuzzy C-means (FCM) is incorporated in [2]. Since the fuzzy weights are informative for the clusters (a microphone can, although dominated by the same source, also contain more noise or interference than another microphone), we also adopted FCM in our approach. This also keeps the comparison between the two methods fair.

In this work, however, we also want to take a look at the distance metric used in the FCM. The standard Euclidean distance was used by Gergen *et al*. In considerable speaker verification implementations, the cosine similarity is used to decide if a new sample is from the same speaker as a previously embedded speaker. This is because the direction and orientation of the embeddings are more discriminative than the distance between embeddings. Therefore, we change the distance metric of the FCM to the cosine distance and compare them to the Euclidean distance metric.

To evaluate these, we proposed the histograms of the direct-to-reverberant and direct-to-reverberant-interference-and-noise ratios (DRR and DRINR) in [11]. Additionally, we will again evaluate the quality of cluster-based speaker separation from [5]. Both methods provide more informative evaluation metrics than trying to generate ground truths since defining this is not straightforward, as in so many other clustering problems.

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

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

$$= \sum_{j=1}^{J} x_{j,m}^{\mathrm{dir}}(n) + x_{j,m}^{\mathrm{rev}}(n) + v_m(n)\,, \tag{2}$$

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. The source signal is further split into the direct path component $x_{j,m}^{\mathrm{dir}}$ and the reflections $x_{j,m}^{\mathrm{rev}}$.

In the following, we often use the short-time Fourier domain representation of the signal for processing, which will be denoted with capital letters:

$$Y_m(l,k) = \mathrm{STFT}\big[y_m(n)\big]\,, \tag{3}$$

where $l$ is the STFT time index and $k$ is the frequency bin.

## 3 Cluster Features

There are three major categories of feature types on which clustering has been performed. We term the first set the geometry-based features (GBFs), the second the signal-based features (SBFs) and the third set the source-dependant latent features (SDLFs). For the GBFs, information about the closeness of microphones is gathered from the signal. For example, in [6] and [8], first the room impulse responses (RIRs) need to be estimated, in order to cluster the different microphone signals. Alternatively, in [4], they use the noise-only periods to compute the magnitude squared coherence (MSC). Under the assumption of diffuse noise, the MSC contains the distance information between the microphones used to cluster the microphones.

The MSC on the signal-active periods is employed in [3] in order to cluster the microphones: the correlation is higher between microphone signals close to each other and lessens the farther they are apart. This is an example of what we term SBFs. The signal-active MSC correlates both the room environment (via the room impulse responses (RIRs)) and the dry speech signal.

Another example of SBFs can be found in [7], where they estimate the coherence of the speech and noise at each microphone with the help of voice activity detection (VAD) and codebooks.

SDLFs argue that signals from microphones close to the same source will generate similar latent features. Additionally, they should be very different from the signals captured by microphones closer to other sources or microphones far from all sources (which are more dominated by the background noise and reflections). An example here is [15], where Gergen *et al*. proposed to compute Mod-MFCC features for each microphone signal and cluster based on these features.

A federated learning based approach to get SDLFs is proposed in [9]. Here an auto-encoder is trained to recreate clean music and speech signals. Then the bottleneck layer of the auto-encoder is changed to have random parameters. These model parameters are then sent to all the nodes. As is done in federated learning, all nodes will then train their network on the specific data of that node, which will in this case overfit on that data. The bottleneck layer weight changes of each node are used as cluster features, since those should be similar for microphone signals dominated by the same speaker.

Although the latter two methods try to focus on the source-specific characteristics, there will always be some influence of the room characteristics. Therefore, we propose to use speaker verification networks to generate source-specific features. This is motivated by improved performances of the deep speaker embedding networks in recent years [16]. These networks are trained to generate the same embedding for the same speaker, irrespective of the (room) environments.

Additionally, the embeddings should be sufficiently unique in order to discriminate between different speakers. Thus these features should also be a good

indication of source dominance at a microphone and therefore good features to cluster on.

For our study, we will compare the Mod-MFCC features and our proposed speaker verification-based features, since both are SDLFs and utilise the same scheme to cluster and process the features. More, we limit ourselves to two methods since we want to focus on the performance of the embedding features under different conditions with increasing difficulty. Comparing more clustering features would certainly be interesting, but is kept for future work.

### 3.1 MFCC-based Features

We will first discuss the modulated Mel-frequency cepstral coefficients (Mod-MFCC) based features utilised in [2, 15]. These features will form the baseline to compare with the speaker embedding features. The hand-engineered Mod-MFCC-based features are two $\mathcal{N}$-dimensional cepstral modulation ratios (CMR) and one $\mathcal{N}$-dimensional averaged modulation amplitude (AMA), with $\mathcal{N}$ the number of considered cepstrum bins.

We briefly describe how to calculate these features [2], later referred to as $\boldsymbol{\mathcal{F}}^{\text{MFCC}}$. First, the MFCC, $Y_{\text{MFCC}}(\eta, k)$ are computed from the STFTs in (3). Here $\eta$ is the cepstral index. Cepstral mean subtraction (CMS) is applied. In cases where the room impulse response (RIR) is constant over time (or at least slowly varying), CMS reduces the effect of reverberations, resulting in features that better capture the speech structure [17] [18]:

$$\tilde{Y}_{\text{MFCC}}(\eta, k) = Y_{\text{MFCC}}(\eta, k) - \frac{1}{K}\sum_{k=0}^{K-1} Y_{\text{MFCC}}(\eta, k)\,. \quad (4)$$

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

$$Y_{\text{Mod-MFCC}}(\kappa, \eta, \lambda) = \sum_{l=0}^{L-1} \tilde{Y}_{\text{MFCC}}(\eta, \lambda Q + l) e^{-j2\pi l\kappa/L}\,, \quad (5)$$

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

$$\hat{Y}_{\text{Mod-MFCC}}(\kappa, \eta) = \sum_{\lambda=0}^{\Lambda-1} |Y_{\text{Mod-MFCC}}(\kappa, \eta, \lambda)|\,. \quad (6)$$

Then the Cepstral Modulation Ratios (CMR) features and averaged modulation amplitude (AMA) features are defined as:

$$\text{CRM}_{\kappa_1|\kappa_2}(\eta) = \frac{\sum_{\kappa=\kappa_1}^{\kappa_2} \hat{Y}_{\text{Mod-MFCC}}(\kappa, \eta)}{(\kappa_2 - \kappa_1 + 1)\hat{Y}_{\text{Mod-MFCC}}(0, \eta)}\,. \quad (7)$$

$$\text{AMA}(\eta) = \frac{1}{L/2+1}\sum_{\kappa=0}^{L/2} \hat{Y}_{\text{Mod-MFCC}}(\kappa, \eta)\,. \quad (8)$$

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

### 3.2 Speaker Verification Features

We propose to use embeddings generated by speaker verification networks as an alternative to the handcrafted Mod-MFCC-based features. We hypothesise that these embeddings lead to more robust clustering features for our goal of clustering microphones around speech sources. This is because, in speaker verification tasks, embeddings are used to test if two audio utterances are spoken by the same person. Speaker embeddings can be compared using their corresponding similarity metric, which depends on 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, microphones dominated by the same speaker should, similarly, yield embeddings that are nearly identical. For ad-hoc microphone clustering, we utilise the embeddings for each microphone as input features, $\boldsymbol{\mathcal{F}}^{\text{SpVer}}$, for the clustering algorithm directly.

The embeddings are generated by the recent Emphasized Channel Attention, Propagation and Aggregation Time Delay Neural Network (ECAPA-TDNN) [12]. ECAPA-TDNN improves upon the popular x-vector architecture [19] 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) [20] 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) [21] softmax loss function. This enables us to use the cosine similarity as the similarity metric to compare two speaker embeddings. We use the same training procedure as described in the accompanying paper [12].

## 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 [22], 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 Sec. 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 Algorithm

The FCM algorithm minimises the following weighed least-squared error function [23]:

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

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

The minimisation of (9) is accomplished by iteratively updating the cluster centres and FMVs with the following functions:

$$\boldsymbol{\mathcal{C}}_c = \frac{\sum_{m=0}^{M-1} \mu_{m,c}^{\alpha} \boldsymbol{\mathcal{F}}_m}{\sum_{m=0}^{M-1} \mu_{m,c}^{\alpha}} \tag{10}$$

$$\mu_{m,c} = \left( \sum_{\tilde{c}=0}^{C-1} \left( \frac{\delta(\boldsymbol{\mathcal{F}}_m, \boldsymbol{\mathcal{C}}_c)}{\delta(\boldsymbol{\mathcal{F}}_m, \boldsymbol{\mathcal{C}}_{\tilde{c}})} \right)^{2/(\alpha-1)} \right)^{-1} \tag{11}$$

### 4.2 Distance metrics

Gergen *et al*. used the standard Euclidean distance metric:

$$\delta_{\text{Euclid}}(\boldsymbol{\mathcal{F}}_m, \boldsymbol{\mathcal{C}}_c) = ||\boldsymbol{\mathcal{F}}_m - \boldsymbol{\mathcal{C}}_c||_2^2 \tag{12}$$

with $\|.\|_2$ is the $\ell_2$ norm of a vector.

Inspired by the similarity score in speaker verification, we propose to use the cosine distance instead:

$$\delta_{\text{Cos}}(\boldsymbol{\mathcal{F}}_m, \boldsymbol{\mathcal{C}}_c) = 1 - \frac{\boldsymbol{\mathcal{F}}_m^T \boldsymbol{\mathcal{C}}_c}{||\boldsymbol{\mathcal{F}}_m||_2 ||\boldsymbol{\mathcal{C}}_c||_2} \tag{13}$$

The main difference with the cosine distance is that only the *direction and orientation* of the vectors influence how closely related the vectors are. Those are exactly the most discriminating part of the embedding features because of the AAM loss function. The magnitude plays no role anymore in the cosine distance.

## 5 Cluster-Based Source Separation

The separation framework used in this work is *identical* to that described in [22, 5]. More sophisticated separation methods, that use cross channel correlations to statistically optimise the separation are available [24, 25]. Nevertheless, the relatively simple framework still succeeds to improve the separation quality. Additionally, the gradual improvements of the framework are accomplished by combining all the information from FCM clusters (both hard and fuzzy labels). This makes it so the quality of the speaker separation is readily correlated with the cluster quality, which makes it a good evaluation tool. Regardless, improving upon the separation method is within our list of future work.

The steps of the separation framework are as follows: firstly, we obtain an initial estimate of the target source in each cluster by means of time-frequency masking (Sec. 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 (Sec. 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) (Sec. 5.3). As the last step, the improved source estimates are used to compute a postfilter (Sec. 5.4).

### 5.1 Initial source estimation

For the initial estimate, we deploy time-frequency (T-F) masking, where the masks, $\mathcal{M}(l,k)$, indicate target dominated bins with one, and noise or interference dominated bins with zero. Masking uses the empirically examined assumption that the localised sources are approximately W-disjoint in their STFT representation [26]. The mask is then element-wise multiplied with the microphone signals to get the source estimate $\widehat{X}_m^{\text{Mask}}(l,k)$:

$$\widehat{X}_m^{\text{Mask}}(l,k) = \mathcal{M}(l,k)\; Y_m(l,k) \tag{14}$$

In order to get 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 the amplitude and have a rough indication of where source dominated T-F bins are. Additionally, we can also compare it to a microphone from the background cluster, which helps to suppress reverberations.

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:

$$Y_c^{\text{ref}}(l,k) = Y_m(l,k) \text{ if } \mu_{m,c} > \mu_{\bar{m},c}\ , \quad \forall \bar{m} \in \{0,\cdots,M-1\}, \bar{m} \neq m \tag{15}$$

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:

$$\mathcal{M}_c(l,k) = \begin{cases} 1 & |Y_c^{\text{ref}}(l,k)| > \dfrac{1}{B}\sum_{b=l-B+1}^{l} |Y_{\bar{c}}^{\text{ref}}(b,k)|, \\ & \forall \bar{c} \in \{0,\cdots,C-1\}, \bar{c} \neq c \\ 0 & \text{else}\,. \end{cases} \tag{16}$$

Here, we also have to introduce the averaging parameter $B$, which is not needed in conventional binary masks. However, this is necessary for the ASN setting, since the 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.

### 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 gotten by a simple delay and sum beamformer. For this, we only include microphones that are close enough to the source. In contrast to compact microphone arrays, the inclusion of more microphones does not necessarily improve the separation capability of the beamformer [4]. Only microphones with enough target energy should be considered.

To get the selection of microphones that are part of a cluster, we transform the fuzzy clusters into hard partitionings. A microphone $m$ is allocated to cluster $c$ if:

$$\mu_{m,c} > \mu_{m,\bar{c}}, \quad \forall \bar{c} \in \{0,\cdots,C-1\}, \quad \bar{c} \neq c\,. \tag{17}$$

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 those. To that end, 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 $\widehat{\tau}_{m,c}$ with respect to the estimated at the reference microphone of cluster $c$ is then computed from these estimates and is obtained by simple correlation analysis. Time-alignment is then performed to the unprocessed microphone signals $y_{m,c}$ and then averaged together, yielding the DSB output for cluster $c$:

$$\widehat{x}_c^{\text{DSB}}(n) = \frac{1}{M_c}\sum_m y_{m,c}(n - \widehat{\tau}_{m,c})\,, \tag{18}$$

Note that the original microphone signals, and not the masked signals, are beamformed together since we do not want the distortions caused by the masks in the beamformer output.

### 5.3 fuzzy value aware delay and sum beamforming

As an extension to the DSB, [5] proposed a fuzzy membership value aware DSB. This better exploits the information given by the FCM, where ideally, the microphone closest to the source will have the greatest

FMV for that source cluster, while microphones further from the source will have a slightly lower FMV for that source cluster and a higher FMV to the background cluster (or interfering source cluster).

$$\widehat{x}_c^{\text{FMVA-DSB}}(n) = \frac{1}{\sum_m \mu_{m,c}} \sum_m \mu_{m,c}\, y_{m,c}(n - \widehat{\tau}_{m,c})\,. \tag{19}$$

Note that in (19), the $y_{m,c}$'s are still only the signals that are part of the hard clustering.

### 5.4 postfilter

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 remove with simple beamforming. The postfilter computed on the output of the DSBs as follows:

$$\mathcal{M}_c^{\text{Post}}(l,k) = \begin{cases} 1 & |\widehat{X}_c(l,k)| > \frac{1}{B}\sum_{b=l-B+1}^{l} |\widehat{X}_{\bar{c}}(b,k)|, \\ & \forall \bar{c} \in \{0,\cdots,C-1\}, \bar{c} \neq c \\ 0 & \text{else}\,. \end{cases} \tag{20}$$

and can be applied to the *beamformed* signal in a similar manner to (14)

## 6 Experiments

Through the simulation of different microphone and source positions in realistic room acoustics, we will try to confirm our hypothesis that speaker embeddings count as good cluster features and are more robust than the Mod-MFCC based features. We increase the difficulty of the scenarios in order to get a better understanding of the capabilities of the method, which will be described in Sec. 6.1.

We evaluate the quality of the clusters by assessing the clustered microphones themselves and by the subsequent cluster based source separation. The quality of the fuzzy weights will also indirectly be evaluated by the FMVA-DSB, as well as through the choice of the reference microphone for each cluster. These metrics will be defined in Sec. 6.3

### 6.1 evaluation setup - SINS database

For the evaluations, we make use of the realistic room impulse responses (RIRs) available in the SINS databset, which was also used in [10]. The database is the simulated version of the apartment layout in [13], where CATT-Acoustic with cone-tracing [27] performed the auralisation. This is an important step towards validation of the system in real world settings 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.

The LibriSpeech clean speech [28] database is preprocessed in order to be used as dry sources for the simulations. The processing is similar to what [14] proposed: signals of 10s are selected from the train-clean-100 LibriSpeech subset. A voice activity detector ensures that there is speech in those 10s segments. The recordings of different speakers have different amplitudes. We do not normalise the speech to be equally loud, which results in a possible combination of speakers where one is four times louder than the other. Note that for the training of the ECAPA-TDNN, a different database is used. The ECAPA-TDNN is trained on the voxceleb 1&2 database [29], where audio of around 7250 celebrities is scraped from YouTube.

The apartment room from [13] is depicted in Fig. 1. There is a big living area (with an open kitchen), a bedroom, a bathroom, a toilet and a hall. The total floor area is $50m^2$. We split the scenarios into two sets based on the inter-source distance. This makes it possible to evaluate the behaviour of the system under these different conditions and see the possible deterioration under more challenging conditions.

The first set of scenarios aims to have easily interpretable results and is a direct parallel to what we previously did in the shoe box acoustics. The scenario only selects sources and microphones from within the living (and kitchen) area. Here we take one source in the left half of the room, and another one in the right half of the room. For maximum interpretability, we avoid scenarios where the critical distances of the sources can overlap for this set. Then 16 microphones will be placed in the room, from which 3 microphones are forced to be picked from within the critical distance of each source, and the other $16-3J$ microphones are chosen at random.

We note that the database consists of four-element microphone array nodes. Since we are only interested in distributed microphones, we only pick one microphone from the same node. We do, however, pick a random microphone from the microphone array in order to increase the diversity in the scenarios.

The second set of scenarios will increase the difficulty of microphone-cluster assignment by bringing the sources closer to each other. The sources are at most separated by three times the critical distance of that room, while the minimum distance is defined by the dataset: 0.4m. Since the critical distance of this room

is equal to 0.68m, the interfering source can lie in the critical distance of the other. Here it will be interesting to see if the system completely breaks down, or if the clusters can still be understood. Additionally, the source separation method will be thoroughly tested in these scenarios and might indicate that a better scheme is necessary.

Another point we want to evaluate is how the segment length for feature extraction influences the resulting clusters. For that, we will revert to the first set of scenarios, to reduce the influence of other factors. We will take 4sec as our baseline since that was previously taken. We will evaluate segment lengths of $4; 2; 1$ and 0.5 seconds. Ideally, the clusters should not change.

Lastly, we also want to incorporate a known speaker embedding into the clustering algorithm. For this, we generate a scenario where another speaker is constantly active, while the known speaker becomes active after some time and later becomes inactive again. Since we are looking for the speaker, we initialise one cluster centre as the precomputed speaker embedding of the known speaker. While the target source is inactive, we would ideally have an empty cluster, while the cluster should contain microphones while the source is active.

### 6.2 parameters

All audio signals are sampled at 16kHz. The von Hann window of length 512 samples (32ms) and window shift of 160 samples (10ms) is applied for the STFTs. The MFCC parameters are: $L = 16$ and $Q = 8$. After throwing away the zeroth MFCC-bin, since we don't want to cluster based on energy, we take the first $\mathcal{N} = 13$ MFCC features, resulting in a 39-dimensional feature vector $\boldsymbol{\mathcal{F}}^{\text{MFCC}}$. In contrast, the speaker verification feature $\boldsymbol{\mathcal{F}}^{\text{SpVer}}$ length is 192. However, a longer feature vector does not necessarily lead to more informative features, which is the case for the Mod-MFCC features. The quality of the features is more important than the dimension of the feature vector. The averaging factor $B$ for the mask computation in (16) and (20) is set to 5. For clustering, we use the fuzzy C-means python package [30].

### 6.3 evaluation metrics

Defining a good quality measure for good ad-hoc microphone clusters is not straight forward. Ground truths are not readily available, making it difficult to define appropriate performance metrics. Others have proposed ways 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: the reflections can be stronger in some parts of the room than others. this makes it so the desired radius within the microphones should also be dependent on the room position. Additionally, it should also depend on how close the interfering sources are. In such cases, it might additionally be beneficial to have non-circular boundary regions and include more microphones along the opposite side from the interferer. Using the oracle RIRs ground truths does solve the problem of creating non-circular boundaries, but is not easily adaptable to include a background cluster.

Generating ground truths also have the disadvantage of making hard cut-offs. However, some wrong decisions are worse than others (is a microphone position right outside the cut-off or extremely close to the interferer instead?). The normalised cluster-centroid–to–source distance metric used in [15] 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.

To solve this, we proposed 3 metrics ourselves in [11], which we will discuss in Sec. 6.3.1. Another way to evaluate the cluster quality, is by evaluating the subsequent tasks. In [2], they evaluate the results based on gender classification. In this paper, we evaluate the clusters based on objective speaker separation measures, which will be explained in Sec. 6.3.2

#### *6.3.1 cluster metrics*

We suggested 3 alternative metrics to intuitively evaluate the clustering performance. The goal is to have an indication of whether the clustering favours microphones with a strong direct-path component and a good signal to interference and noise ratio – which is optimal for the subsequent inference or enhancement stages. To do so, we compute (i) the *distribution* of the direct-to-reverberant ratio (DRR) and (ii) the *distribution* of the direct-to-reverberant, interference, and noise ratio (DRINR) for each microphone allocated to a *speech-source* clusters. These metrics are defined as follows:

$$\text{DRINR} = \frac{\sum_n (x_{c,m}^{\text{dir}}(n))^2}{\sum_n (y_m(n) - x_{c,m}^{\text{dir}}(n))^2} \quad \text{and} \tag{21}$$

$$\text{DRR} = \frac{\sum_n (x_{c,m}^{\text{dir}}(n))^2}{\sum_n (x_{c,m}^{\text{rev}}(n))^2}\,. \tag{22}$$

A distribution centred around high DRRs and DRINRs values means that te clustering is good at

selecting only those with relevant information about the speaker of that cluster. We also want to indicate the amount of spatial diversity available for subsequent tasks, by also reporting (iii) the average number of microphones allocated to a speech cluster.

*6.3.2 source separation metrics*

We additionally evaluate the cluster quality based on the subsequent source separation metrics. Good clusters will lead to good source separation. The quality of the separation thus tells something about the clusters. 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. A decent SIR can be achieved by suppressing the interfering source completely, while only keeping a small portion of the target speech. This would however lead to unintelligible, poor-quality speech. Therefore we also use the perceptual quality (PESQ: perceptual evaluation of speech quality [32]) and intelligibility (STOI: short-time objective intelligibility [33]) metrics.

## 7 Results and Discussion

### 7.1 first set of scenarios

For the first set of scenarios, the results are plotted in Fig. 2 to 4. 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 [11]. Here, we see fairly similar patterns: in Fig. 2, 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 separation methods and evaluation metrics. Fig. 3b and 4a also show that the Mod-MFCC features tend to include more microphones with relatively low source dominance. In contrast, the speaker embeddings present higher main lobes, suggesting that they are able to find slightly more useful microphones.

Making use of the cosine distance, the Mod-MFCC features do improve greatly and the separation performance becomes closer to the separation performance of the speaker embedding based features. However, as is evident in the DRR and DRINR distributions in **??** and Fig. 4b, the speaker embeddings in combination with the cosine distance select even more source dominated. This superior microphone selection also translates into better speaker separation, visible in Fig. 2.

In general, we can state that for these scenarios, the cosine distance is better than the euclidean distance, and the speaker embeddings outperform the Mod-MFCC based features. The combination of cosine distance and Mod-MFCC based features can come closer to the performance of the combination of speaker verification and euclidean distance but is still slightly beat.

### 7.2 second set of scenarios

The results for this set scenario can be seen in Fig. 5 to 7. Bringing the sources closer, unsurprisingly, makes the clustering harder. All speaker separation metrics are lower than for the first scenario. SIRs are even frequently below 0 dB, which means that the interferer is picked up more than the actual target. However, since the source can have different signal amplitudes, it is possible that for these close sources, the whole room is dominated by one of the sources, making it nearly impossible to separate those with the chosen separation scheme. Therefore, it is important to investigate more robust separation methods.

One interesting observation on the speaker separation metrics is that the cosine distance seems to give a great improvement over Euclidean distance. This is most outspoken for the initial mask, which indicates that the choice of reference microphone is superior with cosine distance. Here, in contrast to where sources are always far apart, the combination of Mod-MFCC with the cosine distance does outperform the combination of speaker embeddings with the Euclidean distance. However, when both features utilise the cosine distance to optimise the clustering, the speaker verification features do come out on top.

The histograms in Fig. 6 and 7 tell a similar story, where the speaker verification features select more useful microphones while lowering the number of non-target dominated microphones. Nevertheless, it is intriguing to zoom in on the region of less than ideal microphones (-10dB DRR and lower) in Fig. 6a. There, the DRR histogram would suggest that the Mod-MFCC features are more robust than the speaker embeddings. However, when looking at the DRINR distribution in Fig. 7a, the conclusions seem to be reversed. This indicates that the speaker embeddings are better at incorporating information about the interferer speaker, rather than only the distance of a microphone to the target speaker.

### 7.3 segment length

Fig. 8 to 10 show the impact of shortening the evaluation length given to the feature extractors, for the same scenarios as Sec. 7.1.

For the Mod-MFCC features, the clusters degrade under shorter segment lengths, certainly for lengths of 1s and 0.5s. In contrast, for the embeddings, the segment length seems to only have a limited impact on the

clusters, even for the short length of 0.5s. This is visible in the 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 outspoken. This shift also gets more prominent for shorter evaluation lengths.

The same effect is visible in the speaker separation metrics in Fig. 8: the performance of the Mod-MFCC based features starts dropping with lower evaluation lengths. At 0.5s, the drop is quite significant. In contrast, the performance of the embedding based speaker separation stays quite consistent.

This shows that the speaker embeddings can either be computed on short segments. This can be utilised to lower the computational complexity of the feature extraction. Also, the shorter lengths make the method capable to adapt the clusters quickly to changing situations.

### 7.4 known speaker embedding

We only show one example for this scenario. Therefore, we pick the speaker verification features with together with the cosine distance. Since we want to track the changes in the clustering, we make use of the shorter evaluation segments. A length of 2s is chosen. Fig. 1 shows the results for this scenario.

In the first and last part (Fig. 1a and 1c, where only the interferer is active, there is an empty source cluster. 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, FCM does identify the source and clusters microphones around it, which can be seen in Fig. 1b.

Note that here, we only evaluated using one scenario. However, in order to perform a full blown evaluation, more considerations to take into account. For example, if the sources are closer to each other, or the interferer is louder than the target, the embeddings generated by the microphone signals will differ substantially from that of the know embedding. This would possibly result in an empty source cluster, even when the target source is active. This would require careful design changes, which are kept for future work. Nonetheless, here, we only wanted to demonstrate that this method works in a simple case.

## 8 Conclusions

In this paper, we showed the robustness of the speaker embedding features to cluster ad-hoc distributed microphones under realistic and challenging simulations. They are robust against the target sources being close to each other and hardly degrade when context information gets shorter. Even with an evaluation length of 0.5 seconds, the clusters stay similar to evaluation lengths of 4 seconds. We further showed that speaker embeddings are better features for clustering and that the cosine distance outperforms the Euclidean distance when it comes to determining the similarity between two microphones. This was shown using the DRR and DRINR distributions, and by showing the performance on the subsequent simple cluster based separation step. Our future work includes investigating how to extend the separation framework to, for example, utilise the target and noise coherences between microphones with the help of the information present in the clusters. Also, to further investigate the use of known embeddings withing the clustering algorithm.

**Acknowledgements**
The authors thank Rainer Marting from Ruhr University Bochum (RUB) for the discussion. Also thanks to Alexander Bohlender from Ghent University (UGent) for the quick feedback on the paper draft.

**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 on reasonable request. The RIR dataset should be asked to the owners of the SINS dataset.

**Ethics approval and consent to participate**
Not applicable.

**Competing interests**
We corresponded with and got the SINS dataset from Rainer Martin, who is a Guest Editor of this EUSIPCO special issue "Signal Processing and Machine Learning for Speech and Audio in Acoustic Sensor Networks"

**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 the SINS simulated RIRs and code in order to use and plot this. N.M. conceptualised the idea to use speaker embeddings and gave comprehensive feedback on the wringing.

**Author details**
[1]IDLab, Department of Electronics and Information Systems, Ghent University - imec, Ghent, Belgium. [2]Institute of Communication Acoustics, Ruhr-Universität Bochum, Bochum, Germany.

**References**

1. A. Bertrand. Applications and trends in wireless acoustic sensor networks: A signal processing perspective. In *2011 18th IEEE symposium on communications and vehicular technology in the Benelux (SCVT)*, pages 1–6. IEEE, 2011. 1
2. S. Gergen, A. Nagathil, and R. Martin. Classification of reverberant audio signals using clustered ad hoc distributed microphones. *Signal Processing*, 107:21–32, 2015. 2, 4, 5, 8
3. A. J. Muñoz-Montoro, P. Vera-Candeas, and M. G. Christensen. A coherence-based clustering method for multichannel speech enhancement in wireless acoustic sensor networks. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pages 1130–1134. IEEE, 2021. 2, 3, 8

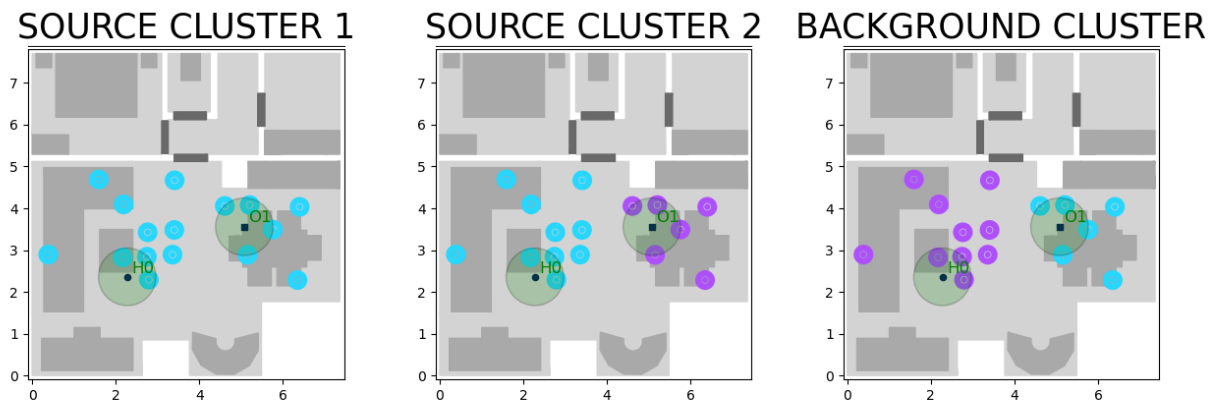
4. I. Himawan, I. McCowan, and S. Sridharan. Clustered blind beamforming from ad-hoc microphone arrays. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):661–676, 2010. 2, 3, 6, 8
5. S. Gergen, R. Martin, and N. Madhu. Source separation by fuzzy-membership value aware beamforming and masking in ad hoc arrays. In *Speech Communication; 13th ITG-Symposium*, pages 1–5. VDE, 2018. 2, 3, 5, 6
6. S. Pasha, Y. X. Zou, and C. Ritz. Forming ad-hoc microphone arrays through clustering of acoustic room impulse responses. In *2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, pages 84–88. IEEE, 2015. 2, 3, 8
7. Yingke Zhao, Jesper Kjær Nielsen, Jingdong Chen, and Mads Græsbøll Christensen. Model-based distributed node clustering and multi-speaker speech presence probability estimation in wireless acoustic sensor networks. *The Journal of the Acoustical Society of America*, 147(6):4189–4201, 2020. 2, 3
8. Matthias Dziubany, Rüdiger Machhamer, Hendrik Laux, Anke Schmeink, Klaus-Uwe Gollmer, Guido Burger, and Guido Dartmann. Machine learning based indoor localization using a representative k-nearest-neighbor classifier on a low-cost iot-hardware. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 2050–2054, 2018. 2, 3
9. Alexandru Nelus, Rene Glitza, and Rainer Martin. Estimation of microphone clusters in acoustic sensor networks using unsupervised federated learning. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 761–765. IEEE, 2021. 2, 3
10. Luca Becker, Alexandru Nelus, Rene Glitza, and Rainer Martin. Accelerated unsupervised clustering in acoustic sensor networks using federated learning and a variational autoencoder. In *2022 International Workshop on Acoustic Signal Enhancement (IWAENC)*, pages 1–5. IEEE, 2022. 2, 7
11. S. Kindt, J. Thienpondt, and N. Madhu. Exploiting speaker embeddings for improved microphone clustering and speech separation in ad-hoc microphone arrays. In *2023 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2023. (ACCEPTED BUT NOT PUBLISHED). 2, 3, 8, 9, 11
12. B. Desplanques, J. Thienpondt, and K. Demuynck. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. In *Interspeech 2020*, pages 3830–3834. International Speech Communication Association (ISCA), 2020. 2, 4, 5
13. Gert Dekkers, Steven Lauwereins, Bart Thoen, Mulu Weldegebreal Adhana, Henk Brouckxon, Bertold Van den Bergh, Toon Van Waterschoot, Bart Vanrumste, Marian Verhelst, and Peter Karsmakers. The sins database for detection of daily activities in a home environment using an acoustic sensor network. *Detection and Classification of Acoustic Scenes and Events 2017*, pages 1–5, 2017. 2, 7
14. Alexandru Nelus, Rene Glitza, and Rainer Martin. Unsupervised clustered federated learning in complex multi-source acoustic environments. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pages 1115–1119. IEEE, 2021. 2, 7
15. S. Gergen and R. Martin. Estimating source dominated microphone clusters in ad-hoc microphone arrays by fuzzy clustering in the feature space. In *Speech Communication; 12. ITG Symposium*, pages 1–5. VDE, 2016. 3, 4, 8
16. Andrew Brown, Jaesung Huh, Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxsrc 2021: The third voxceleb speaker recognition challenge. *arXiv preprint arXiv:2201.04583*, 2022. 3
17. F. Bimbot, J. F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-García, D. Petrovska-Delacrétaz, and D. A. Reynolds. A tutorial on text-independent speaker verification. *EURASIP Journal on Advances in Signal Processing*, 2004(4):1–22, 2004. 4
18. P. N. Garner. Cepstral normalisation and the signal to noise ratio spectrum in automatic speech recognition. *Speech Communication*, 53(8):991–1001, 2011. 4
19. David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333, 2018. 4
20. Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018. 4
21. Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4685–4694, 2019. 5
22. S. Gergen, R. Martin, and N. Madhu. Source separation by feature-based clustering of microphones in ad hoc arrays. In *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, pages 530–534. IEEE, 2018. 5
23. James C Bezdek, Robert Ehrlich, and William Full. Fcm: The fuzzy c-means clustering algorithm. *Computers & geosciences*, 10(2-3):191–203, 1984. 5
24. Shmulik Markovich-Golan, Alexander Bertrand, Marc Moonen, and Sharon Gannot. Optimal distributed minimum-variance beamforming approaches for speech enhancement in wireless acoustic sensor networks. *Signal Processing*, 107:4–20, 2015. 5
25. Dani Cherkassky, Shmulik Markovich-Golan, and Sharon Gannot. Performance analysis of mvdr beamformer in wasn with sampling rate offsets and blind synchronization. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pages 245–249. IEEE, 2015. 5
26. S. Rickard and O. Yilmaz. On the approximate W-disjoint orthogonality of speech. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–529. IEEE, 2002. 6
27. Bengt-Inge Dalenbäck. Tuct v2.0e:1, catt. `http://www.catt.se`, 1999. Accessed: 2019. 7
28. Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE, 2015. 7
29. Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. *Proc. Interspeech 2018*, pages 1086–1090, 2018. 7
30. M. L. D. Dias. Fuzzy $c$-means: An implementation of fuzzy $c$-means clustering algorithm., May 2019. 8
31. E. Vincent, R. Gribonval, and C. Févotte. Performance measurement in blind audio source separation. *IEEE transactions on audio, speech, and language processing*, 14(4):1462–1469, 2006. 9
32. A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra. Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In *IEEE Intl. Conf. on acoustics, speech, and signal processing.*, volume 2, pages 749–752, 2001. 9
33. C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *IEEE Intl. Conf. on acoustics, speech and signal processing*, pages 4214–4217, 2010. 9


**Figures**

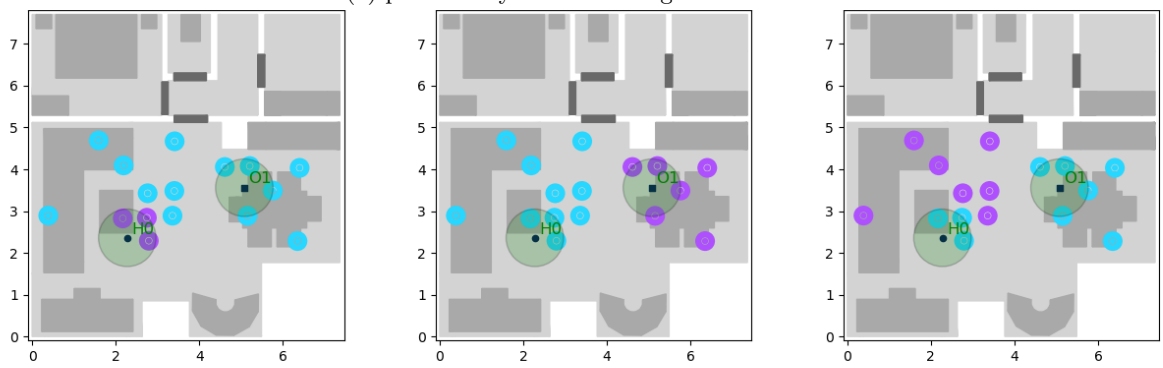
**Tables**

**Additional Files**

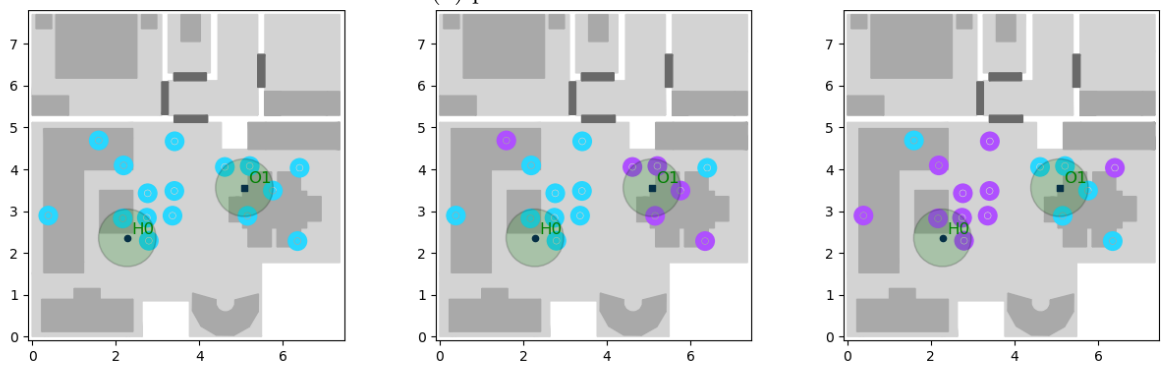
The reference [11], which has been accepted for ICASSP2023




(a) part 1: only the interfering source is active




(b) part 2: both sources active




(c) part 3: only the interfering source is active

Figure 1: Scenario where in the first part, only the second source is active. In the second part, both sources are active. And in the last part, only the second source is active again. In this scenario, we initialise one of the clusters with the known speaker embedding of the first speaker. Therefore, it is able to generate an empty cluster in (a) and (c)

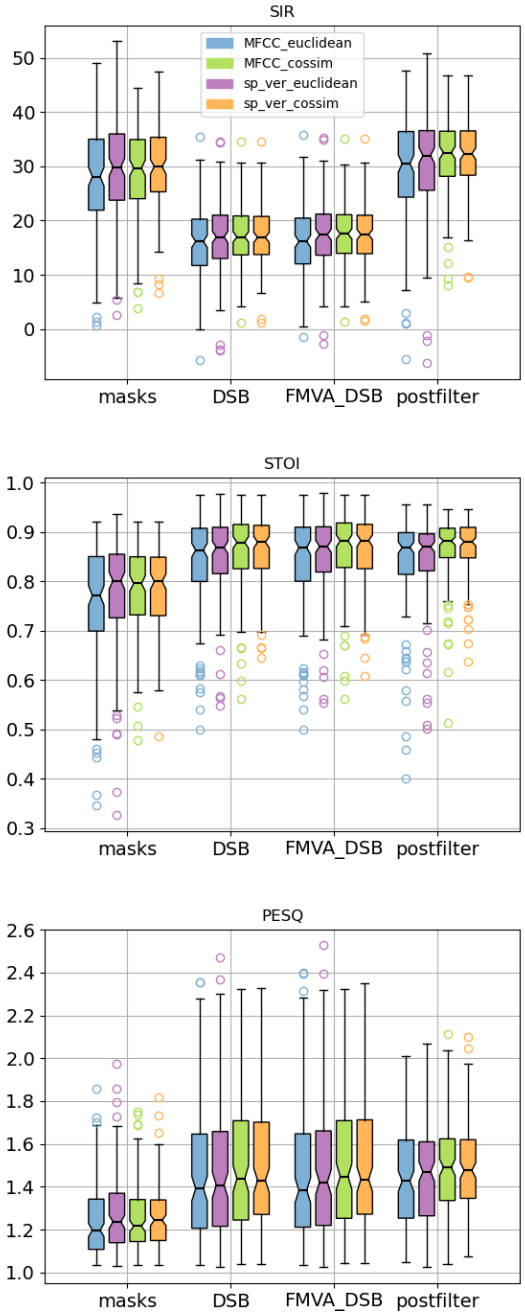
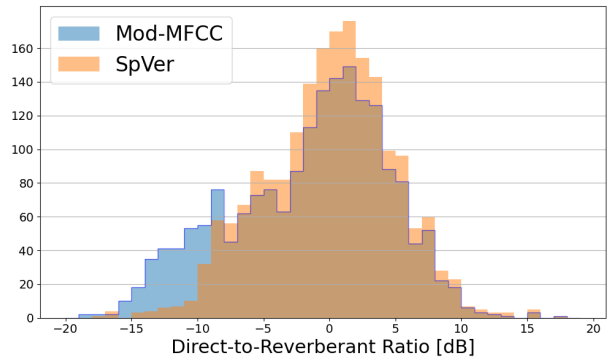
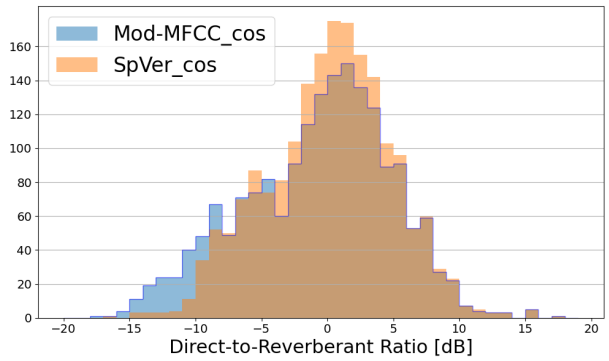



Figure 2: Performance metrics (SIR, PESQ, STOI) showing the separation effectiveness of the cluster feature types (colours) and method (x-axis) for the first set of scenarios, where the sources are always sufficiently far apart.

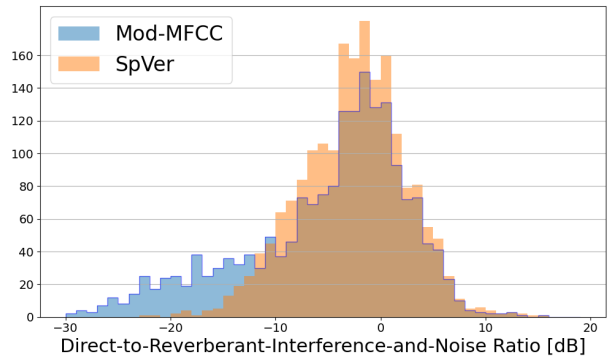



(a) Euclidean distance

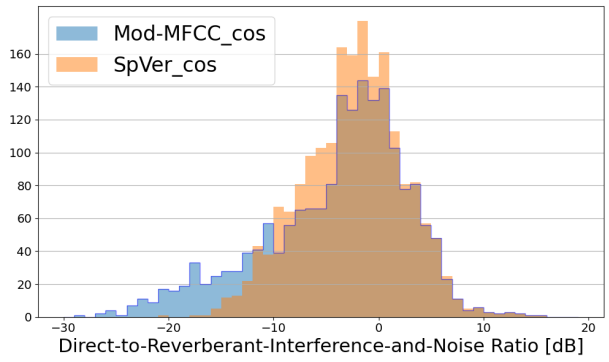



(b) Cosine distance

Figure 3: Histograms of the direct-to-reverberant (DRR) 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.




(a) Euclidean distance




(b) Cosine distance

Figure 4: Histograms of the direct-to-reverberant, interference, and noise ratio (DRINR) 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.

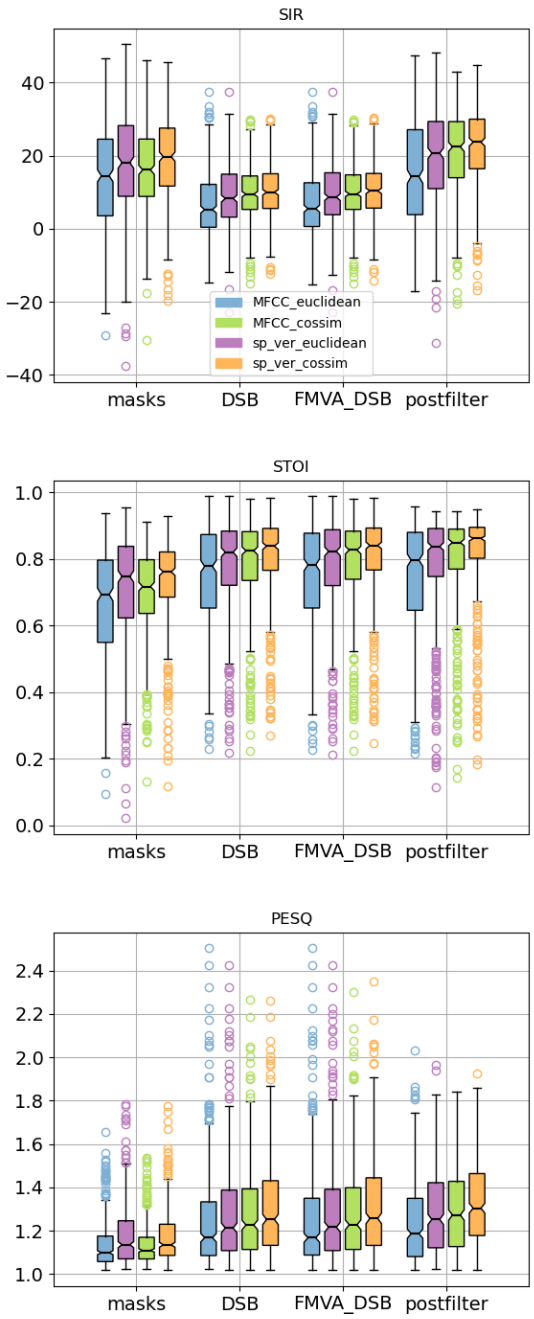
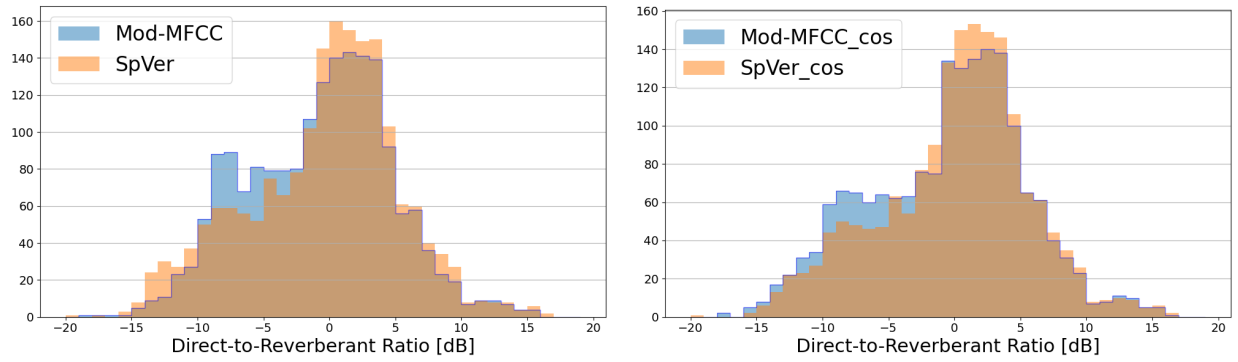
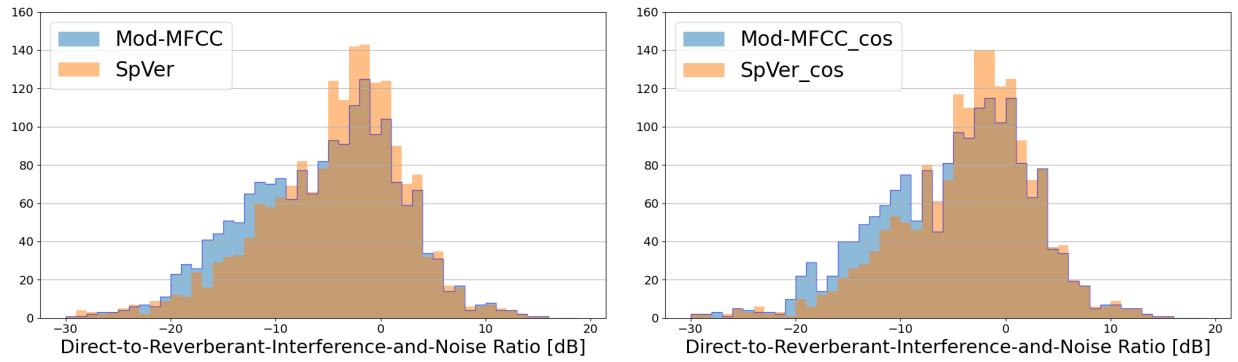



Figure 5: Performance metrics (SIR, PESQ, STOI) showing the separation effectiveness of the cluster feature types (colours) and method (x-axis) for the second set of scenarios, where the sources are maximally separated by three times the critical distance.




(a) Euclidean distance (b) Cosine distance

Figure 6: Histograms of the direct-to-reverberant, interference, and noise ratio (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.




(a) Euclidean distance (b) Cosine distance

Figure 7: Histograms of the direct-to-reverberant, interference, and noise ratio (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.

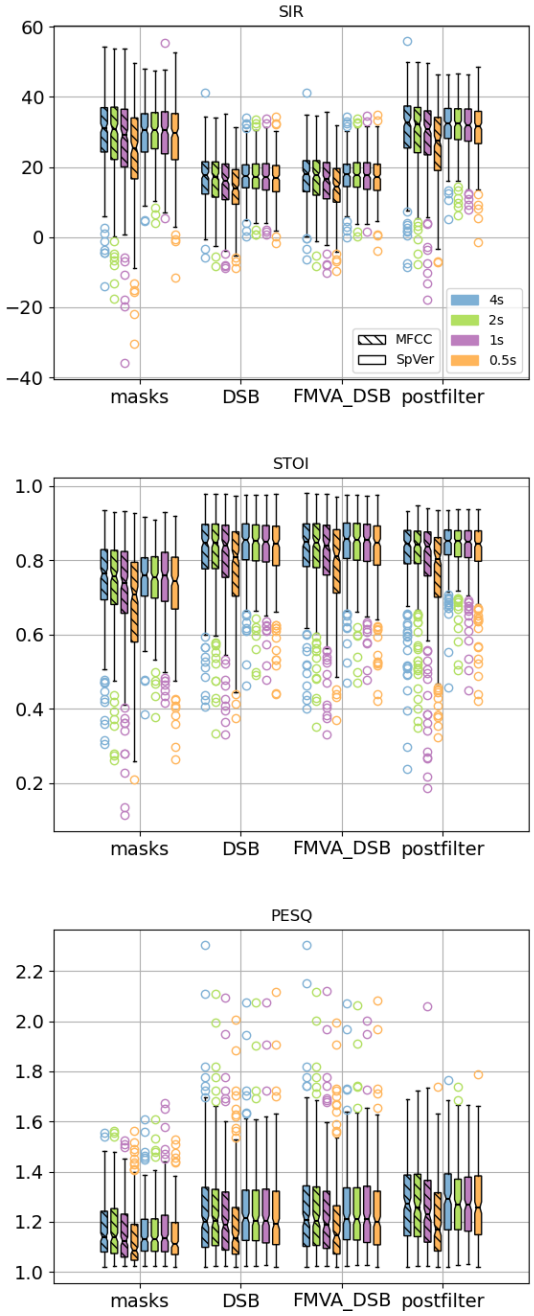
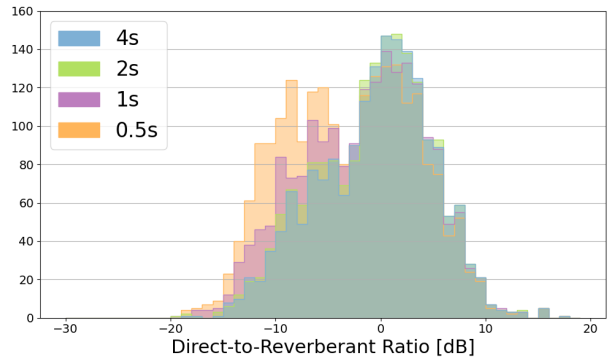
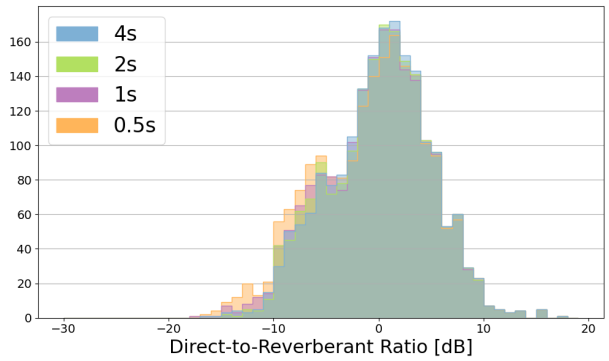



Figure 8: Performance metrics (SIR, PESQ, STOI) showing the separation effectiveness of the cluster feature types (hatches), method (x-axis) and duration (colour) for the first set of scenarios, where the sources are always sufficiently far apart.

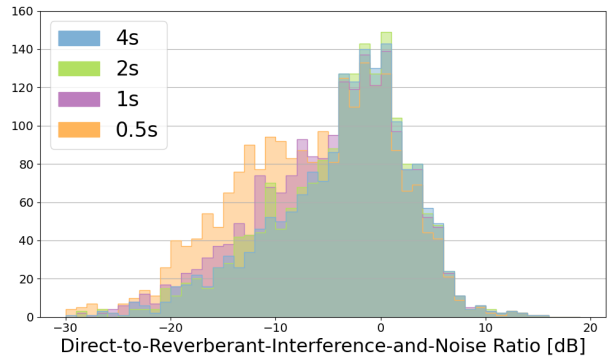



(a) MFCC

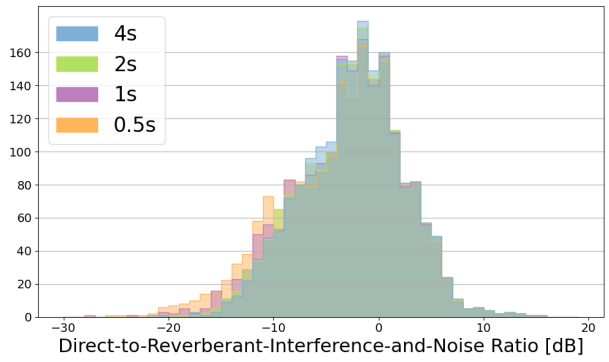



(b) SpVer

Figure 9: Histograms of the DRRs of the (a) Mod-MFCC features (b) and 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.




(a) MFCC




(b) SpVer

Figure 10: Histograms of the DRINRs of the (a) Mod-MFCC features (b) and 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.