DNN-based mask estimation for distributed speech enhancement in spatially unconstrained microphone arrays

Nicolas Furnon, Romain Serizel, Slim Essid, Irina Illina

DNN-BASED MASK ESTIMATION FOR DISTRIBUTED SPEECH ENHANCEMENT IN SPATIALLY UNCONSTRAINED MICROPHONE ARRAYS

Nicolas Furnon, Romain Serizel, Slim Essid, Irina Illina

***Abstract*—Deep neural network (DNN)-based speech enhancement algorithms in microphone arrays have now proven to be efficient solutions to speech understanding and speech recognition in noisy environments. However, in the context of ad-hoc microphone arrays, many challenges remain and raise the need for distributed processing. In this paper, we propose to extend a previously introduced distributed DNN-based time-frequency mask estimation scheme that can efficiently use spatial information in form of so-called compressed signals which are pre-filtered target estimations. We study the performance of this algorithm named Tango under realistic acoustic conditions and investigate practical aspects of its optimal application. We show that the nodes in the microphone array cooperate by taking profit of their spatial coverage in the room. We also propose to use the compressed signals not only to convey the target estimation but also the noise estimation in order to exploit the acoustic diversity recorded throughout the microphone array.**

I. Introduction

SPEECH enhancement aims to recover the clean speech from a noisy signal. It can be used in applications as diverse as automatic speech recognition, hearing aids, and (hand-free) mobile communication. Single-channel speech enhancement, relying on a single microphone signal, can substantially increase the speech quality but the noise reduction is often accompanied by an increase in the speech distortion. Multichannel speech enhancement can overcome this limitation by exploiting the spatial information provided by several microphones. One can distinguish the data-independent multichannel filters [1] from the data-dependent multichannel filters [2]–[5], which depend on the estimation of the statistics of the noisy signal, the noise signal or the target signal. The multichannel Wiener filter (MWF) is a data-dependent multichannel filter, which is optimal in the mean squared error (MSE) sense. It can be extended to the speech distortion weighted multichannel Wiener filter (SDW-MWF) [6] which enables a trade-off between the noise reduction and the speech distortion. Most of these multichannel filters have been developed in constrained microphone arrays, where the number and positions of microphones are fixed and where all the microphones share a common clock. They are called centralized solutions, because a so-called fusion center gathers all the signals of the microphone array.

With the multiplication of embedded microphones in wireless portable devices that surround us, ad-hoc microphone arrays have gained interest [7]. They can be considered as heterogeneous, unconstrained microphone arrays, which are much more flexible and can cover a wider area than traditional microphone arrays. However, the dependency of the centralized approaches on a fusion center makes these solutions too constrained and unrealistic. Many solutions have been proposed to distribute the processing over the whole microphone array in order to get rid of the fusion center, based on a reduction of the transmission costs [8]–[10] or on distributed processing [11]–[14]. Bertrand and Moonen introduced a distributed adaptive node-specific signal estimation (DANSE) algorithm, where each node, instead of sending all its signals to a fusion center, sends only one signal, called compressed signal, to the other nodes, thus reducing the bandwidth cost in addition to cancelling the need of a fusion center [15].

All of these methods rely on the knowledge either of the (relative) acoustic transfer functions, or of the target signals covariance matrices, or both. Recently, deep neural network (DNN)-based solutions have enabled great progress to accurately estimate these parameters, most of the time by predicting time-frequency (TF) masks from a single-channel input [16]–[18]. However, it is also possible to exploit the multichannel information to better estimate these parameters. The spatial information can be explicitly given to a DNN through handcrafted features [19], [20], or implicitly by feeding the DNN either with the multichannel short-time Fourier transform (STFT) signals [21]–[23] or with the multichannel raw waveforms [24], [25].

Although these DNN-based methods lead to promising results and manage to exploit multichannel information, most of them are centralized solutions. Very little work has been published on DNN-based speech enhancement in ad-hoc microphone arrays. Ceolini and Liu [26] introduced a DNN-based method that can process real-time speech enhancement in an ad-hoc microphone array but their solution relies on a centralized minimum variance distortionless response (MVDR) and the DNN is not able to exploit multichannel information. In a previously published paper, we introduced a distributed DNN-based mask estimation that could exploit the multichannel data

N. Furnon, R. Serizel and I. Illina are with the Université de Lorraine, CNRS, Inria, Loria, F-54000 Nancy, France. email: nicolas.furnon@loria.fr

Slim Essid is with the LTCI, Télécom Paris, Institut Polytechnique de Paris, Palaiseau, France.

to better predict the masks [27]. Tested in a simple scenario with two nodes, it outperformed a MWF applied to the nodes separately.

This paper proposes an extended study of our previously introduced speech enhancement scheme [27]. By analysing its performance under various configurations, including real world ones, we confirm that it matches DANSE performance in terms of source to interferences ratio (SIR) and source to artifacts ratio (SAR) even when DANSE is implemented using an oracle voice activity detector (VADOR). We also evaluate in detail the performance under varying SIRs and reverberant conditions and highlight the cooperation of the nodes in the microphone array. This study shows that, depending on the characteristics of the signals captured by the sending and the receiving nodes, sending the so-called compressed signal could be optimized by deciding to send the estimation of either the target or the noise.

Besides, we analyse the performance of the DNNs used in this context. In particular, we investigate the influence of the noise and the spatial diversity between the training and test conditions, looking for a trade-off between performance and robustness to varying scenarios. We also investigate the influence of the quality (in terms of SIR) of the signals used to train the DNNs.

This paper is organized as follows. In Section II, we describe the problem and the multichannel speech enhancement solutions that this paper relies on. In Section III we present our proposal and the challenges that it raises. The experimental setup used to evaluate our proposed solution is described in Section IV. In Sections V and VI, we investigate the performance of the DNNs which have single-channel and multi-channel input. We show in Section VII that sending the noise estimation can lead to improved performance. We conclude the paper in Section VIII.

II. Problem formulation

A. Notations

We consider a fully-connected microphone array with K nodes each having M_k microphones. $M = \sum_{k=1}^{K} M_k$ is the total number of microphones. The signal recorded by the m-th microphone of the k-th node is denoted as $y_{k,m}$. Under the assumption of an additive noise model, in the STFT domain, we have:

$$y_{k,m}(f,t) \;=\; s_{k,m}(f,t) + n_{k,m}(f,t)\,,$$

where $s_{k,m}$ and $n_{k,m}$ denote the speech and noise signals respectively and where f and t denote the frequency and time frame indexes respectively. For the sake of conciseness, we will thereafter omit the frame and frequency indexes unless necessary. The signals from the different channels at node k are stacked into the vector:

$$\mathbf{y}_k \;=\; [y_{k,1}, ..., y_{k,M_k}]^T.$$

All the signals of all nodes are stacked into the vector $\mathbf{y} \;=\; [\mathbf{y}_1^T, ..., \mathbf{y}_K^T]^T$. Similarly, the speech and noise signals are stacked into $\mathbf{s}$ and $\mathbf{n}$. In the following, regular lowercase letters denote scalars; bold lowercase letters indicate vectors and bold uppercase letters indicate matrices.

B. Multichannel Wiener filter

The centralized MWF aims at estimating the speech component s_i of the i-th sensor of the microphone array. The MWF is the optimal filter in the MSE sense, *i.e.* it minimises the MSE between the desired signal s_i and the estimated signal:

$$\mathbf{w}_{\mathrm{MWF}} = \arg\min_{\mathbf{w}} \mathbb{E}\{|s_i - \mathbf{w}^H\mathbf{y}|^2\}. \tag{1}$$

$\mathbb{E}\{\cdot\}$ is the expectation operator and $\cdot^H$ denotes the Hermitian transpose. Solving Eq. (1) yields:

$$\mathbf{w}_{\mathrm{MWF}} = \mathbf{R}_{yy}^{-1}\mathbf{R}_{ys}\mathbf{e}_i\,, \tag{2}$$

where $\mathbf{R}_{yy}$ is the correlation matrix of the input signal, $\mathbf{R}_{ys}$ is the cross-correlation matrix between the input signal and its speech component and $\mathbf{e}_i \in \mathbb{R}^M$ is a vector of zeros with a 1 at the i-th position. Without loss of generality, we will take the channel $i = 1$ as the reference channel in the sequel. The correlation matrices can be obtained as follows:

$$\mathbf{R}_{yy} \;=\; \mathbb{E}\{\mathbf{y}\mathbf{y}^H\} \tag{3}$$

$$\mathbf{R}_{ys} \;=\; \mathbb{E}\{\mathbf{y}\mathbf{s}^H\}\,. \tag{4}$$

Under the assumption that speech and noise are uncorrelated and that the noise is locally stationary, we have:

$$\mathbf{R}_{ys} = \mathbf{R}_{ss} = \mathbb{E}\{\mathbf{s}\mathbf{s}^H\} = \mathbf{R}_{yy} - \mathbf{R}_{nn} \tag{5}$$

where $\mathbf{R}_{nn}$ is the noise correlation matrix:

$$\mathbf{R}_{nn} = \mathbb{E}\{\mathbf{n}\mathbf{n}^H\}\,. \tag{6}$$

Computing these matrices requires the knowledge of noise-only periods and speech-plus-noise periods. This is typically obtained with a voice activity detector (VAD) [6], [15] or a TF mask [16], [26], [28].

A variant of the MWF, called SDW-MWF, was introduced by Doclo *et al.* in order to balance the noise reduction with the speech distortion [6]. Introducing μ, the trade-off parameter between the noise reduction and the speech distortion, the SDW-MWF can be computed as

$$\mathbf{w}_{SDW-MWF} = \left(\mathbf{R}_{ss} + \mu\mathbf{R}_{nn}\right)^{-1}\mathbf{R}_{ss}\mathbf{e}_1\,. \tag{7}$$

Under the assumption that a single speech source is present, Serizel *et al.* proposed a rank-1 approximation of the covariance matrix $\mathbf{R}_{ss}$ based on the generalized eigenvalue decomposition (GEVD) of the matrix pencil $\{\mathbf{R}_{yy}, \mathbf{R}_{nn}\}$ [29]. Combining it with the SDW-MWF extension of the MWF, they designed a filter which proved to be more robust in low signal to noise ratio (SNR) scenarios with a stronger noise reduction [29].

C. Distributed adaptive node-specific signal estimation (DANSE)

The DANSE algorithm is a distributed MWF which aims at estimating the speech component $s_{k,1}$ of the reference microphone of every node k [15], [30], [31]. We still assume that a single speech source is present. In the DANSE algorithm, no fusion center gathers all the signals of all nodes. Instead,

every node k sends only one so-called compressed signal z_k to the other nodes and receives $K-1$ signals from the other nodes. A SDW-MWF is applied to the vector

$$\tilde{\mathbf{y}}_k = \left[\mathbf{y}_k^T,\ \mathbf{z}_{-k}^T\right]^T \quad (8)$$
$$\text{where } \mathbf{z}_{-k} = [z_1, ..., z_{k-1}, z_{k+1}, ..., z_K]^T$$

and outputs the estimated speech signal $\hat{s}_k$ as follows:

$$\hat{s}_k = \mathbf{w}_k^H \tilde{\mathbf{y}}_k \quad (9)$$
$$= \mathbf{w}_{kk}^H \mathbf{y}_k + \mathbf{g}_{-k}^H \mathbf{z}_{-k}\,, \quad (10)$$

where $\mathbf{w}_k = \left[\mathbf{w}_{kk}^T,\ \mathbf{g}_{-k}^T\right]^T$ is the so-called global filter. $\mathbf{w}_{kk}$ and $\mathbf{g}_{-k}$ are filters applied to the noisy signal $\mathbf{y}_k$ and the stacked compressed signals $\mathbf{z}_{-k}$ respectively. Similarly to Equation 7, the global filter can be computed as

$$\mathbf{w}_k = \left(\mathbf{R}_{\tilde{s}\tilde{s},k} + \mu \mathbf{R}_{\tilde{n}\tilde{n},k}\right)^{-1} \mathbf{R}_{\tilde{s}\tilde{s},k} \mathbf{e}_1\,, \quad (11)$$

where $\mathbf{R}_{\tilde{s}\tilde{s},k}$ and $\mathbf{R}_{\tilde{n}\tilde{n},k}$ are estimated from $\tilde{\mathbf{y}}_k$. From Eq. (10), it can be seen that the sub-filter $\mathbf{w}_{kk}$ is applied on the local signals $\mathbf{y}_k$ only, which yields the compressed signal z_k to be sent to the other nodes:

$$z_k = \mathbf{w}_{kk}^H \mathbf{y}_k\,. \quad (12)$$

D. Mask-based multichannel speech enhancement

Originally, the DNN-predicted TF masks were directly applied to the STFT of the noisy signal in order to extract the target speech [32], [33]. This idea continues to be used with a good performance both in the single-channel [34] and the multichannel context [35], but it requires much better TF masks and complex DNN architectures. It also suffers from distortion that can be alleviated by using multichannel filters. In microphone arrays, a common practice is to estimate a TF mask that is not directly applied to the noisy signal, but used to replace the VAD necessary to compute the speech and noise statistics [3]–[5] required by the multichannel filters like MVDR [16], [26] or MWF [28]. Using these TF masks, the speech covariance matrix can be estimated as:

$$\mathbf{R}_{\tilde{s}\tilde{s},k} = \mathbb{E}\{\tilde{\mathbf{s}}_k \tilde{\mathbf{s}}_k^H\} \quad (13)$$

with

$$\tilde{\mathbf{s}}_k = \tilde{\mathbf{m}}_{s,k} \odot \tilde{\mathbf{y}}_k \quad (14)$$

where $\odot$ is the Hadamard product and $\tilde{\mathbf{m}}_{s,k}$ are the stacked TF masks corresponding to the speech components of $\tilde{\mathbf{y}}_k$. To compute the noise covariance matrix, the TF masks $\tilde{\mathbf{m}}_{s,k}$ should be replaced by their complement $\tilde{\mathbf{m}}_{n,k} = 1 - \tilde{\mathbf{m}}_{s,k}$.

To estimate these TF masks, a common practice is to use a DNN which estimates them from a single-channel noisy signal [16], [26], [28]. In a previous work, we showed that we can improve the TF mask prediction, as represented in Figure 1 [27]. In a two-node scenario, we introduced a batch-version of DANSE where at each node, a convolutional recurrent neural network (CRNN) predicted a TF mask out of the reference channel of the node and the compressed signal sent by the other node. To avoid issues related to convergence, we split the iterative process of DANSE into two distinct steps.

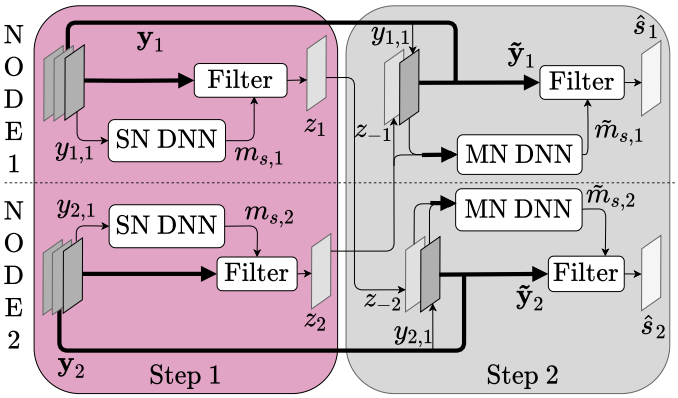


Fig. 1. Illustration of Tango, a two-step speech enhancement algorithm, here applied on two nodes. "SN DNN" and "MN DNN" respectively refer to single-node and multi-node deep neural networks. Bold arrows represent multichannel signals, simple arrows represent single-channel signals.

In a first step (left box of Figure 1), each node processes only local signals to estimate the compressed signal as $z_k = \mathbf{w}_{kk}^H \mathbf{y}_k$ and sends it. In a second step, detailed in Figure 2, each node uses both local and compressed signals to estimate the desired signal. Similarly to the original version of DANSE, the compressed signal is used to compute the speech and noise covariance matrices, but we additionally use it to better predict the TF mask with the multi-node DNN. The pseudo-code of our algorithm is reported in Algorithm 1. We name "Tango" this two-step algorithm based on DANSE. Unless mentioned otherwise, all the filters are SDW-MWF computed with a rank-1 GEVD of the covariance matrices [29] and a trade-off factor $\mu = 1$.

Algorithm 1 Tango algorithm

procedure STEP 1
  for $k = 1 \cdots K$ **do**
    Get the TF mask $m_{s,k}$ with a DNN from $y_{k,1}$
    Compute $\mathbf{R}_{ss,k}$, $\mathbf{R}_{nn,k}$ from $\mathbf{y}_k$ over all samples
    Compute $\mathbf{w}_{kk} = \left(\mathbf{R}_{ss,k} + \mu \mathbf{R}_{nn,k}\right)^{-1} \mathbf{R}_{ss,k} \mathbf{e}_1$
    Compute z_k (Eq. (12))
  end for
end procedure

procedure STEP 2
  for $k = 1 \cdots K$ **do**
    Receive $\mathbf{z}_{-k}$ from the distant nodes
    Get the TF mask $\tilde{m}_{s,k}$ with a DNN from $[y_{k,1}, \mathbf{z}_{-k}^T]$.
    Compute $\mathbf{R}_{\tilde{s}\tilde{s},k}$, $\mathbf{R}_{\tilde{n}\tilde{n},k}$ from $\tilde{\mathbf{y}}_k$ over all samples (Eq. (13))
    Compute $\mathbf{w}_k$ (Eq. (11))
    Compute $\hat{s}_k$ (Eq. (9))
  end for
end procedure

In the rest of the paper, we will refer as *single-node* DNNs to the DNNs which predict a TF mask based on the signal of only one node (*e.g.* the DNN of the first step in Figure 1), and as *multi-node* DNNs to the DNNs which predict a TF mask

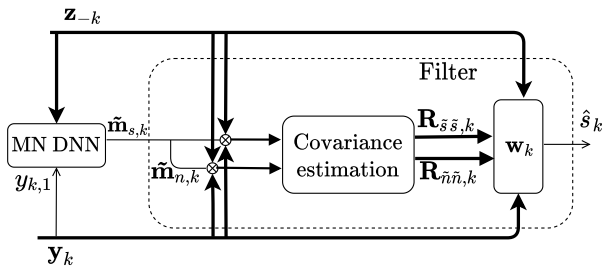


Fig. 2. Detail of the second filtering step. Bold arrows represent multichannel signals, simple ones represent single-channel signals.

based on signals coming from several nodes (*e.g.* the DNN of the second step in Figure 1).

III. Analysis of the Tango algorithm

The solution introduced in our previous work proved that using the compressed signals could help to better estimate the TF masks, thus to increase the speech enhancement performance. We represent in Figure 3 different spectrograms throughout the processing to highlight how useful the compressed signals are in the estimation of the TF masks. As can be seen in Figure 3(f), the TF mask predicted at the second step is less noisy and more accurate than the TF mask estimated at the first step in Figure 3(e), especially at lower frequencies where the different harmonics can be clearly identified. This leads to the filtered signal represented in Figure 3(c), where a higher noise reduction can be observed. In this paper, we propose to extend this solution to more various scenarios, and to evaluate it in challenging scenarios, *e.g.* under high reverberation or in real acoustic conditions. We also address the cases where the signals sent are either the estimation of the target signal or that of the noise, depending on the needs at the receiving node. We propose a detailed analysis of the aspects that have an impact on the final speech enhancement performance and show the benefit of sending the compressed signals among nodes.

A. Single-node networks

In their original version of DANSE, Bertrand and Moonen assumed that all the nodes of the network share the same VAD. That is to say that, when estimating the global filter, at a given time frame, the same binary value was used to estimate the signal statistics for both the reference signal and the compressed signals. This relies on the hypothesis that the speech activity typical variation lasts less than a frame. In our context, as can be seen in Figure 2, TF masks are used and the spectral variation of the speech activity should also be considered (see Equations (13), (14)). Since the signals $\mathbf{y}_k$ of a same device are very similar, the same TF mask is used for all the channels of $\mathbf{y}_k$, but the TF masks of the potentially distant nodes $j \neq k$ should be sent together with $\mathbf{z}_{-k}$ in order to compute $\mathbf{R}_{\tilde{s}\tilde{s},k}$ and $\mathbf{R}_{\tilde{n}\tilde{n},k}$ accurately. This is represented in Figure 4. It translates into a bandwidth overload and we experiment whether we can spare some bandwidth costs by using the TF mask corresponding to $\mathbf{y}_k$ instead of the TF mask corresponding to each $z_{j,\, j\neq k}$.

In addition, we study the influence of the noise diversity in the training data, in a similar manner as Kolbæk *et al.* [36], but where the effects of speech shaped noise (SSN) and real-life noises are analysed separately, so as to distinguish their respective contribution to the training efficiency. SSN is easy to create and overlaps with speech in the STFT domain, representing a cheap but challenging interference, although it is stationary and not representative of real-life noises. On the other hand, real recordings of everyday-life noises are more realistic but require much more time to gather. Our experiment aims at exploring whether the diversity and representativeness brought by the real-life noises can help improving the performance or if training on SSN alone would be sufficient. Likewise, as our proposed solution is evaluated on various spatial configurations, we explore the influence of the spatial configuration it is trained on. We study whether a network should be trained on a specific spatial configuration to achieve high performance on it, or if a trade-off can be found between specificity and generalizability across the spatial configurations at test time.

B. Multi-node networks

The results of our previous paper were obtained on a rather simple dataset, where the spatial configurations were not so diverse and the nodes close to each other. The second part of our work starts by verifying that our previous conclusions generalize well on various scenarios. That is why we evaluate the Tango algorithm on three different spatial configurations. We also analyse its performance when diffuse noise is added on top of the point sources and when real room impulse responses (RIRs) are used to convolve the signals. Besides, we investigate the impact of the reverberation on the performance of the Tango algorithm, as well as the influence of the input SIR. In addition, as the spatial information brought by the compressed signals might be of different interest depending on the receiving node, we propose to repeat in the multi-node context the study relative to the generalizability across spatial configurations. The Tango algorithm is also compared to a state-of-the-art end-to-end multi-channel solution and we show that the SDW-MWF used in Tango brings flexibility that lack in end-to-end solutions.

C. Signals sent among nodes

In a third part, we analyse the importance of adequately selecting the compressed signals sent to the other nodes. Indeed, each node can estimate both the speech and noise components of a noisy signal. In a speech enhancement context, the target signal is the speech signal, but the noise signal may contain very useful information as well, since the MWF also requires the estimation of the noise statistics. Both the speech and noise signals are useful to estimate the speech and noise covariance matrices, and it has also been shown that even a coarse estimation of the noise can help to increase the output performance of a DNN for speech enhancement in the context of automatic speech recognition [20], [21]. An example of this phenomenon is represented in Figure 5 where two nodes see a very different view of the same acoustic scene

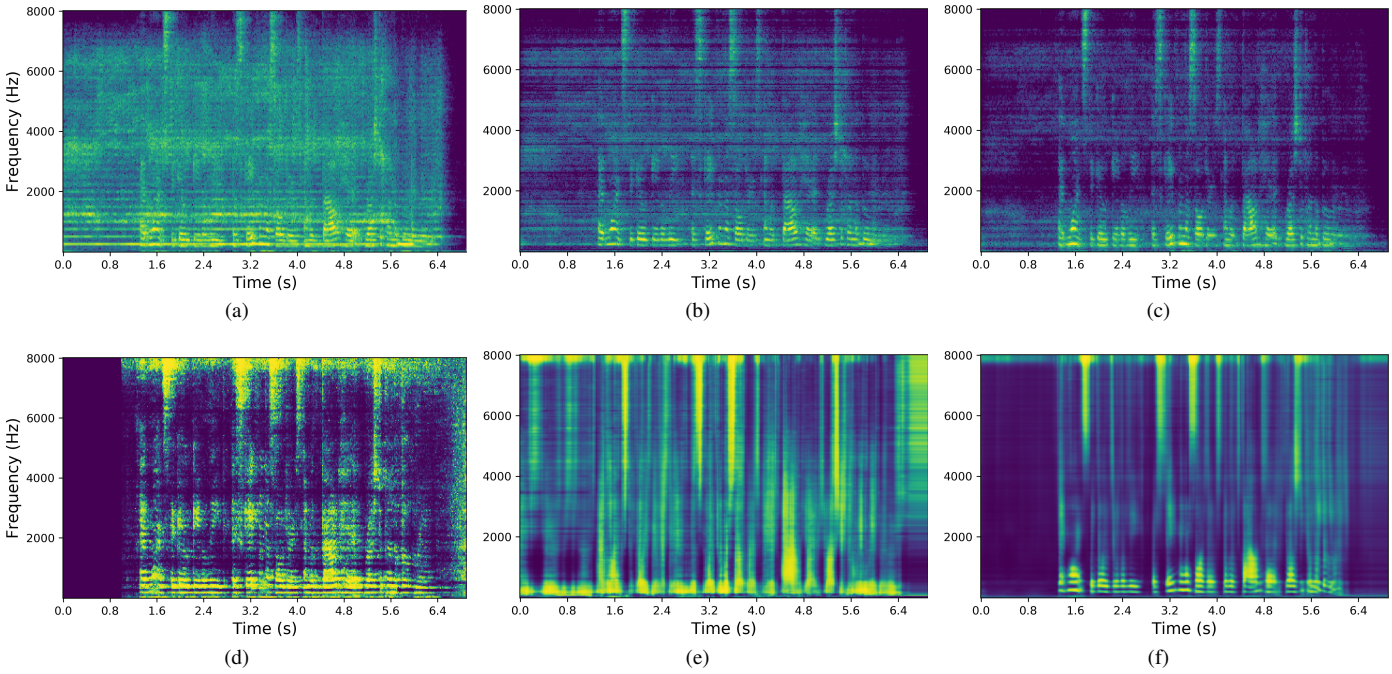
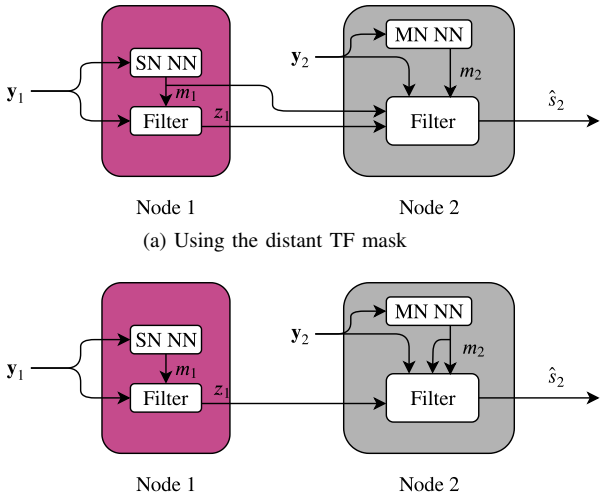


Fig. 3. TF representations of the signals at the first node of a microphone array. (a) Noisy input. (b) Compressed signal sent from node 2. (c) Output enhanced signal. (d) Ideal ratio TF mask corresponding to the input. (e) Mask predicted by the single-node DNN. (f) Mask predicted by the multi-node DNN.



(a) Using the distant TF mask

(b) Using the local TF mask

Fig. 4. Representation of how using the local TF mask can spare some bandwidth cost.

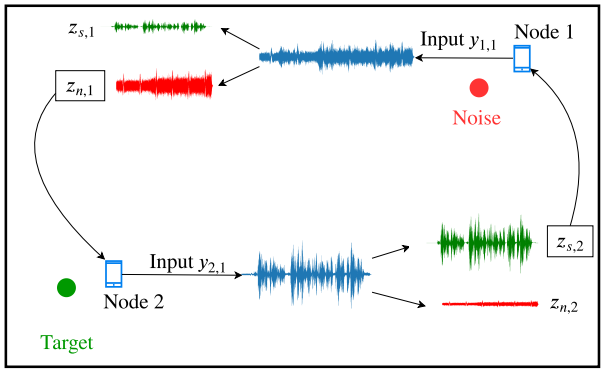


Fig. 5. Example of a situation highlighting the importance of the information provided by the compressed signals. The first node, close to the noise source, can accurately estimate the noise component and send it to the second node. The second node, close to the target source, can accurately estimate the target component and send it to the first node.

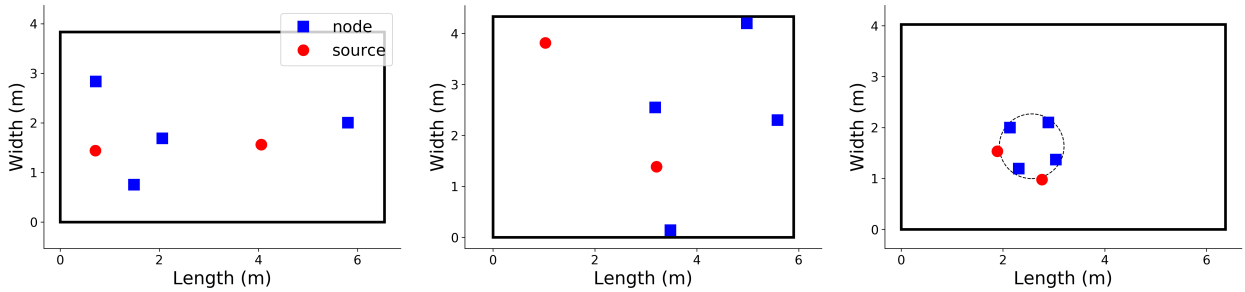
because of their locations in the room. We check which signal (*i.e.* the estimation of the noise or the estimation of the target speech) a given node should send to the other nodes depending on its location in the room.

IV. Setup

A. Datasets with simulated room impulse responses

We create three spatial scenarios with the Python toolbox Pyroomacoustics [37]. An example of each scenario can be seen in Figure 6. Two of these datasets, called *living room* and *meeting room*, aim at simulating the real-life scenarios that correspond to two typical use cases of a living room and a meeting room. To see whether training the DNNs on one generic dataset could generalize well on the test sets of the *living room* and *meeting room*, we create the *random room* configuration, which is less constrained and covers the specific cases of the *living room* and *meeting room*.

In each scenario, shoebox-like rooms are created with a reverberation time (RT) randomly selected between 0.15 s and 0.4 s, the length between 3 m and 8 m, the width between 3 m and 5 m and the height between 2.5 m and 3 m. $K = 4$ recording devices (called *nodes* in the rest of the paper) are simulated, each embedded with four microphones ($M_k = 4 \quad \forall k \in [\![1; K]\!]$). The microphones are on a square at a distance of 5 cm to the node center. Two sources, one target source and one noise source, are added. In each



(a) *Random room* configuration (b) *Living room* configuration (c) *Meeting room* configuration

Fig. 6. 2D representations of the three spatial configurations. The acoustic effect of the table (dashed-line circle in (c)) is not simulated; it is only represented for a better visualization.

scenario, the speech content is taken from the LibriSpeech clean subsets [38]. The noise source can be either SSN or a real recording of everyday-life noises, downloaded from Freesound [39]. In the latter case, the noise files are downloaded by searching for all the files corresponding to a set of keywords[1] and post-processed by hand to discard irrelevant outputs, resample the relevant outputs and remove the silent parts[2]. The noise source signals are amplified by a random gain between -6 dB and 0 dB. After convolution, most of the SNRs lie in the range [-10; +10] dB depending on the node position in the room.

The first scenario, called *random room* (see Figure 6(a)), has very few additional constraints. The two sources and the nodes are randomly placed in the room with the only constraints that they all should be distant of at least 50 cm from each other and from the walls. The nodes are at a random height between 0.7 m and 2 m, as if they were recording devices laid on a piece of furniture, or hearings aids worn by an impaired person. The sources are between 1.20 m and 2 m high, to fit the standard height of most noise sources.

The second scenario, called *living room* (see Figure 6(b)) recreates a situation that could typically happen in a living room with one target speech source and one interference noise source. Three nodes are placed within 50 cm from the walls as if they were on shelves and the fourth device is placed randomly in the room, at 50 cm at least of the walls and the other nodes. All the nodes are at a random height between 0.7 m and 0.95 m. The two sources are also randomly placed in the room at 50 cm at least from the nodes and the walls and at a random height between 1.20 m and 2 m.

The third scenario, called *meeting room* (see Figure 6(c)) simulates a meeting configuration where two people are sitting around a table. One speaker is the target speaker while the second one is considered as an interferent source. The table is circular, with its radius randomly chosen between 0.5 m and 1 m, its height randomly chosen between 0.7 m and 0.8 m and its center randomly placed in the room. The nodes are placed every 90° on the table, at a random distance between 5 cm and 20 cm from the table edge. The two sources are randomly placed around the table within 50 cm from the table edge, at a random height between 1.15 m and 1.3 m, and at 15 cm at least from the walls. The reflection of the table is not simulated[3].

Each dataset is split into a training set containing 10000 samples of 10 s each, a validation set containing 1000 samples of 10 s and a test set containing 1000 samples whose duration range from 6 s to 10 s. The test dataset does not overlap with the training and validation sets in terms of LibriSpeech speakers and Freesound users. To keep a balanced number of samples in all classes of the training set, the noises corresponding to the classes having not enough samples were used for the test set only[4].

B. Dataset with real room impulse responses

In order to evaluate our solution on real measurements, we used the dataset of Corey *et al.* to reproduce real-world conditions [40]. The dataset contains signals recorded in a large conference room, played by 10 sources and recorded by 160 microphones distributed in 4 wearable arrays of 16 microphones and 12 tabletop arrays of 8 microphones. Since the mixtures in the dataset are very challenging cocktail-party scenarios which we do not address in our research, we decided to use the sweep signals that are available in the dataset to compute real RIRs. We reproduced 1000 scenarios by randomly picking two sources among the 10 available and 4 tabletop arrays among the 12 available. Every second microphone in the array was used to compute the RIR, so that a configuration of 4 nodes of 4 microphones each was reproduced, in a similar geometry as in the simulated setup (see Section IV-A). The unconvolved signals reverberated with the real RIRs are the same as the signals used in the simulated experiments (see Section IV-A).

C. Neural network settings

All the signals are sampled at 16000 Hz. The STFT is computed using a Hanning window of 32 ms with an over-

[1]The keywords were `baby`, `blender`, `dishwasher`, `electric shaver`, `toothbrush`, `fan`, `frying`, `printer`, `vacuum cleaner`, `washing machine`, `water`.

[2]The noise dataset is available at https://zenodo.org/record/4019030.

[3]A Python implementation of the code that enabled us to create these datasets is available at https://github.com/nfurnon/disco/tree/master/dataset_generation.

[4]These noises correspond to the classes `baby`, `blender`, `electric shaver`, `toothbrush` and `frying`.

lap of 16 ms. The same CRNN architecture is used in all experiments, for both single-node and multi-node DNNs. The convolutional part is made of three convolutional layers with 32, 64 and 64 filters respectively, with kernel size 3×3 and stride 1×1. Each convolutional layer is followed by a batch normalisation and a maximum-pooling layer of kernel size 4×1 (no pooling over the time axis). The recurrent layer is a 256-unit GRU, followed by a fully-connected layer with a sigmoid activation function in order to map the output of the network between 0 and 1. The networks are trained with the RMSprop optimizer [41]. The input of the models are STFT windows of 21 frames. The compressed signals necessary to train the multi-node CRNN are obtained by applying a SDW-MWF on the mixtures using oracle TF masks. The ground truth targetted by both single-node and multi-node DNNs are the frames of the ideal ratio mask (IRM) corresponding to the 21 frames of the input mixture. The IRM is computed following Equation (15) where $s_{k,1}$ and $n_{k,1}$ are respectively the reverberated target and noise components of the mixture recorded at the first microphone of the k-th node [42].

$$m_k(t,f) = \frac{|s_{k,1}(t,f)|}{|s_{k,1}(t,f)| + |n_{k,1}(t,f)|} \tag{15}$$

The cost function is the MSE between the target IRM and the predicted TF mask, weighted by the spectrogram of the input mixture, in order to take into account the spectral shape of the speech. The cost function can be expressed as :

$$\mathcal{L}(m_k, \hat{m}_k) = \mathbb{E}\{|(m_k - \hat{m}_k) \cdot y_{k,1}|^2\},$$

where m_k and $\hat{m}_k$ are respectively the target and predicted TF masks at node k and $y_{k,1}$ is the mixture recorded by the reference microphone of node k.

D. Performance evaluation

1) Metrics: In the following, all the performance evaluations are quantified based on the SIR and SAR [43], and on the short term objective intelligibility (STOI) [44]. These metrics require a reference signal and it was shown that the SIR and SAR are very sensitive to the chosen reference [45], [46]. Both the source (non-reverberated) and image (reverberated) signals are valid references and quantify differently the performance. Considering the source signal as the reference enables one to keep a constant reference for all sensors despite the diversity of what they capture. However, it does not allow us to distinguish the distortion due to the reverberation from the distortion due to the filter. On the other hand, considering the image signals as references enables one to quantify the effects of the proposed filters only, but the implicit reference of the GEVD filter at a specific node might not be the explicit reference channel of the metric (see Section V in [29]). To cope with this, we quantify the speech enhancement performance with four metrics. The first metric is the difference between the output SIR and the input SIR[5] when the clean (target and noise) image signals are taken as references[6]. We arbitrarily take the first microphone of each node as the reference of this node. It is denoted by $\Delta\text{SIR}_{\text{cnv}}$ where the subscript $_{\text{cnv}}$ means that the reference signals are the convolved signals. The second metric is the SAR where the convolved signals are considered as the references. This metric is denoted as SAR_{cnv}. The third metric is the SAR where the source signals are the references. It is denoted as SAR_{dry} where the subscript $_{\text{dry}}$ means that the reference signals are the source signals. The difference between SAR_{dry} and SAR_{cnv} could be interpreted as the distortion due to the reverberation of the source signals. By keeping both of these metrics, we can quantify both the problems of denoising and of dereverberation. Lastly, the STOI computed with the convolved references is also considered to better quantify the intelligibility of the denoised signals. It is denoted STOI_{cnv}.

2) Signals considered for the evaluation: Depending on the context, we might be interested in having one well-estimated target signal for the whole microphone array, or one well-estimated target signal for each node of the microphone array. In most cases, one signal would be enough for the whole array, but it might require to send this signal to all the other nodes, resulting in a possibly undesired bandwidth overload. The question of a node-specific speech enhancement algorithm issue has also been discussed by Markovich-Golan *et al*. [47]. In our case, we will mainly focus on estimating the best possible signal for the whole array, this is why, unless mentioned otherwise, the results presented in the remainder of the paper represent the average over the whole test set of the performance at the best output node, *i.e.* at the node with the highest output SIR. However, we will also analyse more in detail the behaviour of the proposed solution at the node with the highest and lowest input SIR in Sections VI-G and VII, in order to highlight the cooperation among nodes in the microphone array and the needs of the nodes concerning the compressed signals that they receive. This will then be mentioned explicitly.

V. Analysis of the performance with single-node networks

This section focuses on several factors that impact the performance of the proposed Tango algorithm using masks estimated with single-node DNNs. As described in Section II-D, speech enhancement process is split in two steps and the compressed signals are sent between nodes to compute the filter of the second step, but the same single-node network is used for both steps.

A. Importance of node-specific TF masks

In this section, we investigate in oracle conditions which TF mask should be applied on the compressed signal. To do so, we compare two cases. In the first case, the TF mask of the node sending the signal (called *distant* node) is applied to the compressed signal in order to compute the speech and

[5]Since we do not simulate any microphone noise, the input SIR is equal to the input SNR.

[6]We noticed that the SIR was quite consistent across the reference signals, whether they were the source signals or the image signals.

TABLE I
SPEECH ENHANCEMENT PERFORMANCE IN ORACLE CONDITIONS IN THE RANDOM ROOM CONFIGURATION WHEN APPLYING THE DISTANT OR LOCAL ORACLE TF MASK ON THE COMPRESSED SIGNAL. THE BEST SIGNIFICANT RESULTS ARE IN BOLD.

	$\Delta\text{SIR}_{\text{cnv}}$ (dB)	SAR_{cnv} (dB)	SAR_{dry} (dB)	STOI_{cnv}
local	26.8 $\pm$ 0.4	**10.9 $\pm$ 0.2**	**9.6 $\pm$ 0.2**	**0.89 $\pm$ 0.004**
distant	26.1 $\pm$ 0.4	8.3 $\pm$ 0.2	9.0 $\pm$ 0.2	0.85 $\pm$ 0.004

noise statistics at the second filtering step. In the second step, the TF mask of the receiving node (called *local* node) is used. The results are reported in Table I where the two cases are respectively referred to as *distant* and *local*.

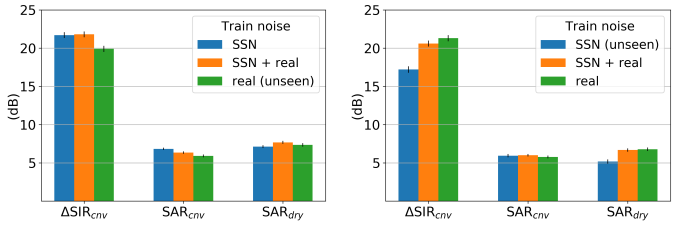
As can be seen in Table I, using the TF mask of the local node instead of the distant node not only limits the bandwidth requirements, but also increases the speech enhancement performance in terms of SAR and STOI without decreasing the SIR. This might come from the fact that the beamformer is robust to small TF mask estimation errors. The drop of SAR_{cnv} and STOI_{cnv} when the distant TF mask is used could be due to the fact that the filtered signal is closer to the reference of the distant nodes than to the reference of the local node. This could decrease the metrics without actually decreasing the performance. The almost equal SAR_{dry} between the two methods seems to confirm this hypothesis. As a conclusion, in the remainder of the paper, the local TF mask will be the one applied on all the compressed signals coming from the other nodes to estimate the signal statistics required by the MWF.

B. Robustness to unseen noise

We trained a model in the random room configuration under three noise conditions. In the first condition, the noise signals are all samples of SSN. In the second condition, the noises are real recordings of everyday-life noises downloaded from Freesound as described in Section IV. In the third condition, the model is trained with half of the signals mixed with SSN noise and the other half mixed with Freesound noises. The three resulting models are tested on noisy signals where the noise is either SSN or a real recording. The corresponding results are represented in Figure 7, where *real* refers to recordings downloaded from Freesound.

The first observation is that the networks trained on a single type of noise are specialized on this noise, *i.e.* they perform better in matched test conditions than in unseen conditions. This is especially true in terms of $\Delta\text{SIR}_{\text{cnv}}$. On the other hand, the network trained on both types of noises performs at least as well as the specialized network. This conclusion is similar to the conclusion of Kolbæk *et al.* [36]. However, because we separately analysed the influence of the SSN and of the real noise, our experiment is additionally able to show that removing the SSN from the training set decreases the generalization capacities of the DNN, in particular on stationary noises. This is confirmed when considering the STOI, which we do not represent here for the sake of conciseness.

As a conclusion to this section, a wider variety of training material leads to a robust network that performs as good as a specialized network in matched conditions, and can maintain performance in unmatched conditions. In the following, since the test set might contain unseen noises during the training, all the networks will be trained on both types of noises described above in order to increase their robustness, but they will be tested on the real noises.



(a) Results on test set with SSN (b) Results on test set with real noise

Fig. 7. Speech enhancement performance of the single-node DNNs in the random room configuration for different training and test noise conditions. The $\Delta\text{SIR}_{\text{cnv}}$ difference in Figure 7b between the two last networks is not statistically significant.

C. Robustness to an unseen spatial configurations

We now consider the impact of the spatial scenario while training the DNN. We compare three DNNs, trained on the signals generated in the three spatial configurations introduced in Section IV-A, and tested on each of these scenarios.

As can be seen in Figure 8, only mildly significant differences can be observed between the three models. One exception can be highlighted, when the DNN trained on the *meeting room* configuration yields the best results, probably because, due to the closeness of some nodes to the noise source, this DNN has seen more challenging scenarios during the training, making it more robust. Apart from this specific single-node scenario, it would not have a big impact to train on one spatial configuration and test on another.

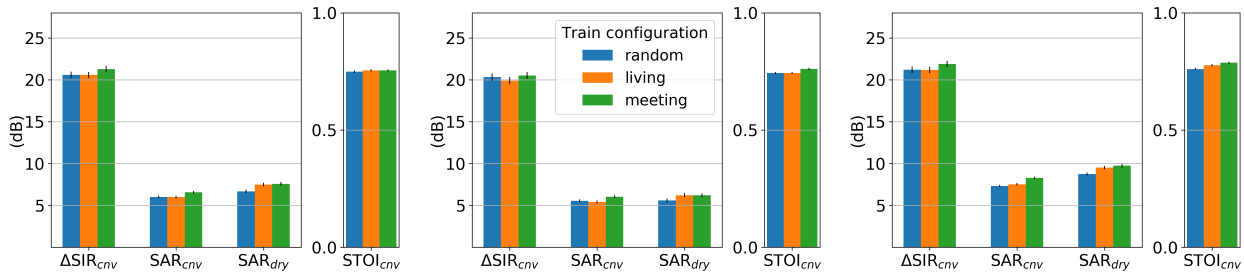
In particular, it is interesting to notice that the SAR_{dry} values are higher in the *meeting room* configuration than in the two other ones. This is because the microphones are close to the target source, which is hence less distorted by the reverberation. This confirms the relevance of the third metric.

VI. ANALYSIS OF THE PERFORMANCE WITH THE MULTI-NODE NETWORKS

We now extend our study to the case where, at the second filtering step, the DNN also receives the signals from the other nodes and uses them as additional input to predict the TF masks. In a similar manner to our previous work, the signals sent are all estimations of the target signal [27].

A. Benefit of using multi-node DNNs

We first study in this section the advantage of using multi-node DNNs over single-node DNNs at the second filtering step. We train a multi-node DNN on each of the spatial configurations introduced in Section IV-A. The compressed signals used to train the DNN are obtained with IRMs, but at test time, the IRMs are replaced by the TF masks estimated by the single-node DNNs. The results are given in Figure 9



(a) Random room (b) Living room (c) Meeting room

Fig. 8. Speech enhancement performance of three single-node networks trained each on a different spatial configuration.

and compared to the oracle case of DANSE, where the signal statistics are computed with an oracle voice activity detector (VADOR). The VADOR is computed from the energy of the convolved target speech signal.

The multi-node DNN brings an $\Delta\text{SIR}_{\text{cnv}}$ improvement of around 3 dB compared to the single-node DNN (see Figure 8), and up to 1.5 dB improvement in terms of SAR. Besides, it increases the performance up to what can be achieved with an oracle VAD in terms of SAR and $\Delta\text{SIR}_{\text{cnv}}$, except on the *meeting room* configuration where the scenario is more challenging. Only the STOI_{cnv} indicates that the oracle knowledge of the VADOR increases the speech intelligibility of the enhanced signal in all configurations, but to a limited extent. Overall, the conclusion of our previous paper that the compressed signals are useful to better predict the TF masks, is confirmed on three real-life scenarios.

B. Influence of the spatial configuration

Given that the compressed signals convey a lot of spatial information, the conclusions of Section V-C might not hold in the multi-node DNN. We repeated the experiments in the multi-node case, where the compressed signals are given at the input of the DNNs. Similarly to the conclusions to Section V-C, there was very little difference across the three DNNs. That is why we will consider only one multi-node neural network in the sequel, the one trained and tested on the *random room*.

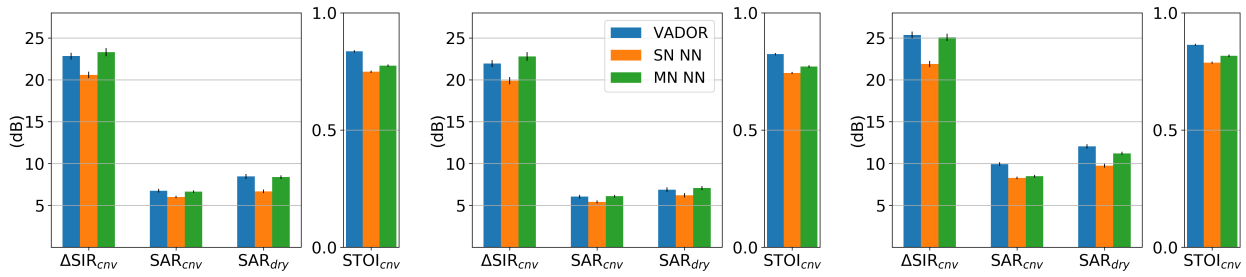
C. Comparison with the state-of-the-art

To further challenge the performance of our solution, we compare it with a multi-channel end-to-end solution called FaSNet [48] which we implemented with the Asteroid toolbox [49]. The original FaSNet architecture is a two-stage filtering where temporal convolutional network (TCN) blocks predict beamforming filters applied on the input channels, based on pair-wise normalized cross-correlations between the input channels. As it reported better results, we replaced the TCN blocks with dual-path RNN (DPRNN) [50]. The modified FaSNet is trained in the *random room* configuration in the same conditions as the CRNN, with the same amount of data, the same signals and the same input SIRs. In each sample, one node (of four microphones) is randomly selected to provide the input mixtures and the output targets. At inference, the trained neural network is applied on all nodes, and the best output is retained to compare the performance of FaSNet with the performance of our proposed Tango algorithm. We compare FaSNet to three versions of the Tango algorithm. The first version uses the oracle TF masks to compute the speech and noise covariance matrices needed at both filtering steps of Tango. The SDW-MWF at both filtering steps is computed based on a rank-1 GEVD of these covariance matrices [29]. The trade-off factor μ (see Equation (7)) is set to 1. The second version follows the same process, but with masks predicted by CRNNs. The third version uses the same CRNNs to predict the masks, but the filter is a SDW-MWF computed from full-rank covariance matrices and with a trade-off factor $\mu = 5$. The results are reported in Table II.

TABLE II
COMPARISON OF DIFFERENT VARIANTS OF THE PROPOSED TANGO WITH FASNET [48]. $\Delta\text{SIR}_{\text{CNV}}$, SAR_{CNV} AND SAR_{DRY} ARE EXPRESSED IN DB.

	$\Delta\text{SIR}_{\text{cnv}}$	SAR_{cnv}	SAR_{dry}	STOI_{cnv}
Tango (oracle) r1-GEVD-SDW-MWF $\mu = 1$	27.1 $\pm$0.4	11.2 $\pm$0.2	9.8 $\pm$0.2	0.90 $\pm$0.003
Tango (CRNN) r1-GEVD-SDW-MWF $\mu = 1$	22.9 $\pm$0.5	6.9 $\pm$0.1	8.5 $\pm$0.2	0.78 $\pm$0.004
Tango (CRNN) SDW-MWF, $\mu = 5$	16.8 $\pm$0.5	13.2 $\pm$0.3	8.8 $\pm$0.3	0.83 $\pm$0.006
FaSNet [48]	17.5 $\pm$0.2	13.8 $\pm$0.2	6.7 $\pm$0.2	0.84 $\pm$0.005

Using masks predicted by the CRNNs rather than oracle masks lowers the performance, but it is interesting to notice how much flexibility using a mask-based approach brings. By choosing the rank of the decomposition of the covariance matrices, and by selecting the value of the trade-off parameter μ, one can tune the speech enhancement and insist either on noise reduction or on speech distortion. FaSNet being trained on a scale-invariant source to distortion ratio (SDR) loss function, it optimizes a compromise between SIR and SAR, that is why both values obtained with FaSNet are rather good and balanced. But the Tango algorithm, based on a SDW-MWF, leaves the freedom to have more noise reduction or speech distortion, which can be decided at run time whereas the output of FaSNet is fixed once the DNN is trained. A



(a) Random room (b) Living room (c) Meeting room

Fig. 9. Speech enhancement performance on the three spatial configurations using an oracle VAD, a single-node DNN and a multi-node DNN at the second filtering step.

good trade-off for example has been found with the full-rank SDW-MWF using $\mu = 5$ (third line in Table II) which performs comparably to FaSNet. Lastly, Tango relies on a much simpler DNN with 0.518 million parameters, whereas FaSNet counts 1.871 million parameters, which makes the DNN in Tango easier to train and quicker at inference time.

D. Performance under diffuse noise

In this section, we study the impact of diffuse noise on the performance of the Tango algorithm. The diffuse noise is simulated by convolving ambient noise with the average of the tail of five RIRs corresponding to five sources randomly placed in the simulated room. The ambient noise files were downloaded from three indoor noise types (`Home`, `Office`, `Library`) of the TUT dataset[7]. The diffuse noise is added in the *random room* configuration at an SNR randomly picked between 0 dB and 20 dB. We represent the speech enhancement results of the Tango algorithm in Figure 10, where the DNNs in Tango were trained in presence of a single point noise source. "PS" refers to the same case as in Section VI-A, where only one point noise source is present. "PS + DN" refers to the case where diffuse noise is present on top of a point noise source. The metrics computed on the input mixture are represented in hatched bars.

Adding diffuse noise lowers the input metrics (SIR_{cnv} and STOI_{cnv}) as well as the output metrics, but to a restricted extent. Lower results are expected because the MWF can be seen as a beamformer [51], which cannot perfectly reduce a noise coming from all directions. However it is worth noticing that, when diffuse noise is added, even if the output STOI_{cnv} decreases, the difference between output and input STOI_{cnv} increases, so the relative performance increases. Overall, this experiment leads to the conclusion that our solution is quite resilient to a diffuse noise field of a moderate intensity.

E. Performance with real RIRs

To further analyse the generalization capacity of the DNNs in Tango, we evaluate our solution on the data obtained with real RIRs described in Section IV-B. The DNNs are trained in the simulated *random room* configuration and evaluated with the signals convolved with real RIRs. The results are given in Table III. They are compared with the performance of the same Tango algorithm in a simulated environment (the same results as in Section VI-A) and with the performance obtained when IRMs are used in the Tango process.

Even with oracle masks, the performance drops when real data is used. This is probably due to the fact that the real evaluation set is more challenging than the simulated one, and the difference between the two oracle systems shows the performance drop that the CRNN-based system cannot avoid. Using a CRNN to predict the masks brings a higher performance loss, especially in terms of SAR. A more detailed analysis showed that the neural network fails to predict accurate masks at higher frequencies, which is detrimental to the speech quality. This could be compensated by incorporating real data in the training set or by using a more complex neural

[7]https://zenodo.org/record/400515

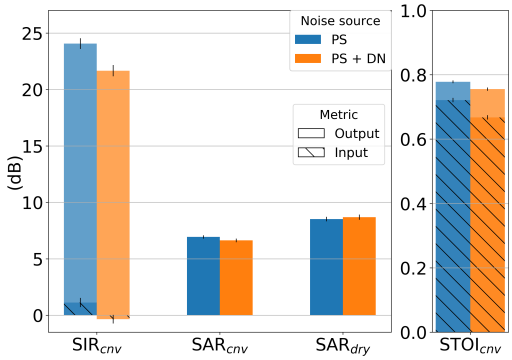
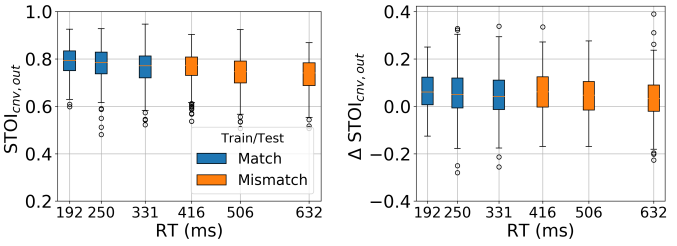


Fig. 10. Speech enhancement performance of the Tango algorithm in the *random room* configuration when diffuse noise is added to the point noise source. "PS" and "DN" respectively refer to a point source and diffuse noise.

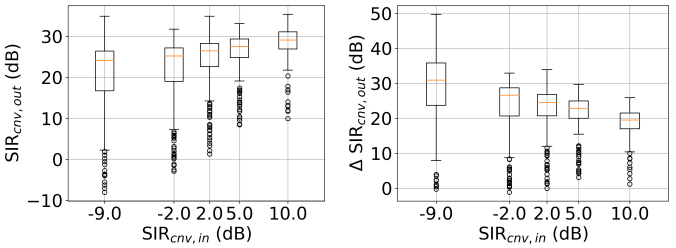
TABLE III
SPEECH ENHANCEMENT PERFORMANCE OF TANGO WITH DNNS TRAINED ON SIMULATED DATA AND EVALUATED WITH SIGNALS CONVOLVED WITH SIMULATED ("SIM" IN THE TABLE) AND REAL ("REAL") RIRS. THE MASKS NEEDED IN TANGO ARE EITHER ORACLE ("IRM") OR PREDICTED BY DNNS ("CRNN").

	$\Delta\text{SIR}_{\text{cnv}}$ (dB)	SAR_{cnv} (dB)	SAR_{dry} (dB)	STOI_{cnv}
IRM sim	26.8 ± 0.4	10.9 ± 0.2	9.6 ± 0.2	0.89 ± 0.004
IRM real	22.7 ± 0.3	7.9 ± 0.2	3.7 ± 0.3	0.79 ± 0.005
CRNN sim	23.3 ± 0.5	6.6 ± 0.2	8.4 ± 0.2	0.77 ± 0.006
CRNN real	19.0 ± 0.4	3.2 ± 0.1	2.9 ± 0.2	0.61 ± 0.008



(a) Output STOI$_{cnv}$ (b) ΔSTOI$_{cnv}$

Fig. 11. Influence of the reverberation time on the performance of Tango in terms of STOI$_{cnv}$ and ΔSTOI$_{cnv}$.



(a) Output SIR$_{cnv}$ (b) ΔSIR$_{cnv}$

Fig. 12. Influence of the input SIR on the performance of Tango in terms of SIR$_{cnv}$ and ΔSIR$_{cnv}$.

network architecture with a more powerful modelling capacity. The influence of the reverberation alone can be analysed more precisely with simulated data in Section VI-F.

F. Influence of the reverberation

To analyse the influence of the reverberation, we evaluate our solution under higher RTs than those which were used in the training set. Figure 11 shows the performance of Tango in terms of STOI$_{cnv}$ (Figure 11(a)) and ΔSTOI$_{cnv}$ (Figure 11(b)) in the *random room* configuration, under several reverberation conditions. In blue, mild reverberation is simulated in the evaluation set, similarly to the training conditions. In orange, stronger reverberation is simulated, which has not been seen during the training. The output STOI$_{cnv}$ slightly decreases when the reverberation gets stronger, but the relative performance in terms of ΔSTOI$_{cnv}$ remains stable. This shows that our solution is resilient over varying reverberation conditions. This conclusion holds when considering the other metrics (SIR, SAR) which we do not represent here for a more concise presentation.

G. Influence of the input SIR

In this section, we analyse the influence of the input SIR on the output performance of the multi-node solution. We show the advantage of using distributed microphone arrays and we highlight the cooperation among nodes of the microphone array with Tango.

First, we study the global behaviour of Tango over the input SIR. The performance is reported in Figure 12. Figure 12(a) shows the absolute performance in terms of SIR$_{cnv}$ whereas Figure 12(b) shows the relative performance in terms of ΔSIR$_{cnv}$. As had already been observed in previous research, the absolute performance increases when the input SIR increases, but the relative performance decreases [52], [53]. These conclusions hold when considering the other metrics (SAR, STOI), which we do not represent here for a more concise presentation. As Kolbæk *et al.* showed, narrowing down the range of the SIR in the training set to have it match the range of the SIR in the test set could improve the performance on the evaluation set [36]. This however requires a priori knowledge on the testing conditions.

Second, we report the performance of our solution at the best input node and at the worst input node of the microphone array in the *random room* configuration. The best (resp. worst) input node is the node with the highest (resp. lowest) input SIR$_{cnv}$. We report these results in Table IV for the single-node solution (indicated by "SN") and for the multi-node solution (indicated by "MN"). To recall, in the single-node solution, the DNN does not have the compressed signals to predict the TF mask at any of both filtering steps. In the multi-node solution, the DNN of the second filtering step has the compressed signals and the local noisy signal to predict the TF mask. The best (resp. worst) input node is indicated with the subscript $_{bi}$ (resp. $_{wi}$) in the table. The distribution of the input SIR$_{cnv}$ corresponding to the best and worst input nodes is represented in Figure 13.

TABLE IV
SPEECH ENHANCEMENT PERFORMANCE OF THE SINGLE-NODE AND MULTI-NODE NETWORKS AT THE BEST AND WORST INPUT NODES OF THE *random room* CONFIGURATION. (Δ)SIR$_{CNV}$, SAR$_{CNV}$ AND SAR$_{DRY}$ ARE EXPRESSED IN DB.

	SIR$_{cnv}$	ΔSIR$_{cnv}$	SAR$_{cnv}$	SAR$_{dry}$	STOI$_{cnv}$
SN$_{bi}$	18.7 $\pm$ 0.6	16.1 $\pm$ 0.5	5.8 $\pm$ 0.2	5.8 $\pm$ 0.3	0.77 $\pm$ 0.005
SN$_{wi}$	14.2 $\pm$ 0.7	16.6 $\pm$ 0.6	3.4 $\pm$ 0.2	3.6 $\pm$ 0.3	0.70 $\pm$0.006
MN$_{bi}$	20.5 $\pm$ 0.7	17.9 $\pm$ 0.6	6.4 $\pm$ 0.2	7.4 $\pm$ 0.3	0.78 $\pm$0.005
MN$_{wi}$	18.1 $\pm$ 0.7	20.5 $\pm$ 0.6	4.2 $\pm$ 0.2	5.8 $\pm$ 0.3	0.74 $\pm$0.006

As can be seen in Table IV, even in the single-node case, the performance is relatively good at the best input node. The ΔSIR$_{cnv}$ is similar to the one at the worst input node, but the output SIR$_{cnv}$ is higher. Using multi-node DNNs at this best

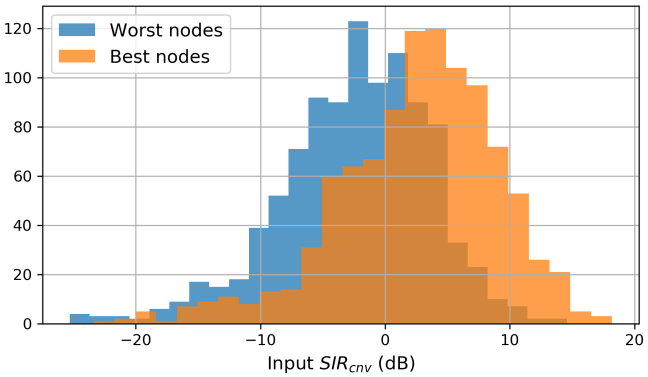


Fig. 13. Histogram of the input SIR$_{cnv}$ at the best input nodes and at the worst input nodes.

input node does improve the final performance, but in a lesser extent than at the worst input node. This is especially true when considering the $\Delta\text{SIR}_{\text{cnv}}$ which increases of almost 4 dB at the worst input node but only 1.8 dB at the best input node. This reduces considerably the discrepancy of output SIR_{cnv} across the whole microphone array. The same observation hold when considering the STOI_{cnv}. It shows that the nodes cooperate and that the DNN on the worst input node is able to exploit the information coming from the other nodes. At the worst input node, the benefit of our method is twofold: the compressed signals come from nodes with a higher SIR and additionally, they are already filtered with a well-predicted TF mask.

As a counterpart, this also probably means that the compressed signals sent by the worst input nodes are not so useful. However, these nodes are the closest to the noise source, so the network could predict quite well the TF mask corresponding to the noise source. Sending the noise estimation as the compressed signals could improve the overall performance. This is what we propose to analyse in Section VII.

VII. Exchanging signals between nodes

In this section, we focus on the compressed signal that is sent from one node to the others. In previous versions of DANSE, only the target estimation was sent [10], [27], [31], [54]. However, as depicted in Figure 5, the noise estimation can also provide useful information, so we propose to compare in which conditions which signal estimation should be sent. To do so, we train three multi-node DNNs. The first one is the DNN that had been used for the previous experiments, and which had as input the target estimations, denoted z_s, coming from the distant nodes, together with the noisy signal at the reference channel. The second DNN has the noise estimations z_n sent from the distant nodes together with the noisy signal at the reference channel. The third DNN has both target speech and noise estimations together with the noisy signal at the reference channel as input. Each of these networks is tested in conditions matching its training conditions, and the results are represented in Figure 14 for the *random room* configuration. We represent the results obtained at the best output nodes (*i.e.* the result is the average over all the filtered signals obtained at the nodes with the highest output SIR_{cnv}), at the best input node (average over the filtered signals obtained at the nodes with the highest input SIR_{cnv}, and at the worst input node (average over the filtered signals obtained at the nodes with the lowest input SIR_{cnv})).

At the best output node (Figure 14a), sending the one or the other compressed signal does not make any difference. At the best input node (Figure 14b), although the differences are not significant, the $\Delta\text{SIR}_{\text{cnv}}$ indicates that this node could benefit from receiving the compressed noise estimation rather than the compressed target estimation. Likewise, using the compressed noise estimation at the worst input node (Figure 14c) leads to worse results, since the worst input node already has good insights on the noise signal and needs an estimation of the target signal, which it can poorly estimate in its own.

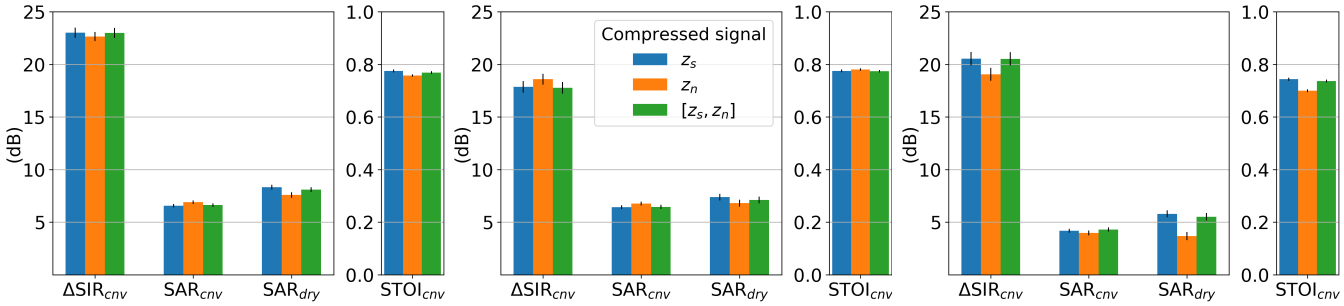
Sending both the target and the noise estimations seems to be very similar to sending only the target estimation. It looks like it does not benefit from the noise estimation at the best input node. However, the significance of the results allows us only to conclude that sending both estimations is not worse than sending either of both. Given the relatively simple architecture of the network, it could also be that sending both signals from all nodes represents an overload of data for the DNN. Carefully selecting either z_s or z_n at the input of the DNN might offer a solution to have the best of both worlds, while alleviating the bandwidth requirements.

Hence, depending on the application, if the aim of the speech enhancement challenge is to have the one best signal for the whole microphone array, then sending only z_s is enough. If each node should have its own estimated signal, as discussed in [47], then depending on the node and its input SIR, a decision has to be taken whether the target or the noise estimation is of greater relevance. Sending both could is an interesting option but it means sending twice more data.

TABLE V
Difference of performance between the first and second filtering steps at the best input node and best output node in the *random room* configuration. $(\Delta)\text{SIR}_{\text{CNV}}$, SAR_{CNV} and SAR_{DRY} are expressed in dB.

	SIR_{cnv}	$\Delta\text{SIR}_{\text{cnv}}$	SAR_{cnv}	SAR_{dry}	STOI_{cnv}
S1_{bi}	17.8 ± 0.4	15.2 ± 0.4	7.4 ± 0.2	7.6 ± 0.2	0.80 ±0.004
S1_{bo}	19.4 ± 0.4	17.6 ± 0.3	7.6 ± 0.1	8.0 ± 0.2	0.80 ±0.004
S2_{bi}	20.5 ± 0.7	17.9 ± 0.5	6.3 ± 0.2	7.4 ± 0.3	0.78 ±0.005
S2_{bo}	24.2 ± 0.5	23.3 ± 0.5	6.6 ± 0.2	8.4 ± 0.2	0.77 ±0.006

Lastly, it is worth noting that the nodes with the best output signal are not always the nodes with the best input signal. The performance of the two filtering steps described in Figure 1 at the best input nodes and at the best output nodes is given in Table V. The first (resp. second) filtering step is mentioned as S1 (resp. S2) and the best input node (resp. best output node) is indicated with the subscript $_{\text{bi}}$ (resp. $_{\text{bo}}$). Even at the first filtering step, where the spatial information is not yet shared, the best input nodes are not always the best output nodes, but the difference of performance between the two types of nodes is quite low. The best output nodes benefit more from the second filtering step than the best input nodes. This is because the performance at the second filtering step (where the multi-node DNNs are used) depends a lot on the compressed signals, which are in general very well estimated by the best input nodes. These compressed signals are received by the other nodes which can benefit from their accuracy and estimate the best output signal. Interestingly, the input SIR_{cnv} of the best output nodes of the second filtering step (equal to 0.9 dB) is lower than the input SIR_{cnv} of the best output nodes of the first filtering step (equal to 1.8 dB). It means that some nodes with a lower input SIR_{cnv} become the nodes with the best overall performance thanks to the information shared across the microphone array. This phenomenon highlights the cooperation among nodes in the proposed algorithm.



(a) Best output node (b) Best input node (c) Worst input node

Fig. 14. Speech enhancement performance of multi-node DNNs trained and tested with different compressed signals in the *random room* configuration.

VIII. Conclusion

We introduced and extended Tango, a DNN-based distributed multichannel speech enhancement methodology which operates in spatially unconstrained microphone arrays. It was evaluated on a large variety of real-life scenarios which proved the efficiency of this solution that performs comparably well to state-of-the-art end-to-end solutions. An evaluation on real data suggested the need of an improved DNN or of adapted training strategies for greater generalization capacities. It was shown that this method is robust to mismatches between the training and test conditions of the DNN. We showed that the nodes with the lowest input SIR benefit the most from the cooperation across the microphone array and we gave insights on the potential benefit of sending the noise estimation rather than the target estimation. An interesting direction of research would be to better select the signals that are needed, either before or after sending them as compressed signals, *e.g.* with attention mechanisms.

Acknowledgment

This work was made with the support of the French National Research Agency, in the framework of the project DiSCogs (ANR-17-CE23-0026-01). Experiments presented in this paper were partially out using the Grid5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000).

References

[1] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP magazine*, vol. 5, no. 2, pp. 4–24, 1988.

[2] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.

[3] O. L. Frost, "An algorithm for linearly constrained adaptive array processing," *Proceedings of the IEEE*, vol. 60, no. 8, pp. 926–935, 1972.

[4] *Optimum Waveform Estimation*. John Wiley and Sons, Ltd, 2002, ch. 6, pp. 428–709.

[5] S. Doclo and M. Moonen, "GSVD-based optimal filtering for single and multimicrophone speech enhancement," *IEEE Transactions on Signal Processing*, vol. 50, no. 9, pp. 2230–2244, 2002.

[6] S. Doclo, A. Spriet, J. Wouters, and M. Moonen, "Frequency-domain criterion for the speech distortion weighted multichannel Wiener filter for robust noise reduction," *Speech Communication*, vol. 49, no. 7-8, pp. 636–656, 2007.

[7] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, "A consolidated perspective on multimicrophone speech enhancement and source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 4, pp. 692–730, 2017.

[8] O. Roy and M. Vetterli, "Rate-constrained collaborative noise reduction for wireless hearing aids," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 645–657, Feb 2009.

[9] J. Zhang, R. Heusdens, and R. C. Hendriks, "Rate-distributed spatial filtering based noise reduction in wireless acoustic sensor networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 11, pp. 2015–2026, Nov 2018.

[10] A. Bertrand and M. Moonen, "Efficient sensor subset selection and link failure response for linear MMSE signal estimation in wireless sensor networks," in *18th European Signal Processing Conference*, Aug 2010, pp. 1092–1096.

[11] Y. Zeng and R. C. Hendriks, "Distributed delay and sum beamformer for speech enhancement via randomized gossip," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 1, pp. 260–273, 2013.

[12] M. Zheng, M. Goldenbaum, S. Stanczak, and H. Yu, "Fast average consensus in clustered wireless sensor networks by superposition gossiping," *IEEE Wireless Communications and Networking Conference, WCNC*, pp. 1982–1987, 04 2012.

[13] G. Zhang and R. Heusdens, "Distributed optimization using the primal-dual method of multipliers," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 173–187, March 2018.

[14] A. I. Koutrouvelis, T. W. Sherson, R. Heusdens, and R. C. Hendriks, "A low-cost robust distributed linearly constrained beamformer for wireless acoustic sensor networks with arbitrary topology," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 8, pp. 1434–1448, Aug 2018.

[15] A. Bertrand and M. Moonen, "Distributed adaptive node-specific signal estimation in fully connected sensor networks — Part I: Sequential node updating," pp. 5277–5291, Oct 2010.

[16] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2016-May, 2016, pp. 196–200.

[17] Z.-Q. Wang and D. Wang, "Mask weighted STFT ratios for relative transfer function estimation and its application to robust ASR," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5619–5623.

[18] H. Erdogan, J. R. Hershey, S. Watanabe, M. I. Mandel, and J. Le Roux, "Improved MVDR beamforming using single-channel mask prediction networks." in *Interspeech*, 2016, pp. 1981–1985.

[19] Y. Jiang, D. Wang, R. Liu, and Z. Feng, "Binaural classification for reverberant speech segregation using deep neural networks," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 22, no. 12, pp. 2112–2121, 2014.

[20] S. Araki, T. Hayashi, M. Delcroix, M. Fujimoto, K. Takeda, and T. Nakatani, "Exploring multi-channel features for denoising-autoencoder-based speech enhancement," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 116–120.

[21] L. Perotin, R. Serizel, E. Vincent, and A. Guérin, "Multichannel speech separation with recurrent neural networks from high-order ambisonics recordings," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 36–40.

[22] T. Yoshioka, H. Erdogan, Z. Chen, and F. Alleva, "Multi-microphone neural speech separation for far-field multi-talker speech recognition,"

in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5739–5743.

[23] S. Chakrabarty and E. A. P. Habets, "Time-Frequency Masking Based Online Multi-Channel Speech Enhancement With Convolutional Recurrent Neural Networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 1–1, 2019.

[24] T. N. Sainath, R. J. Weiss, K. W. Wilson, B. Li, A. Narayanan, E. Variani, M. Bacchiani, I. Shafran, A. Senior, K. Chin, A. Misra, and C. Kim, "Multichannel signal processing with deep neural networks for automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 965–979, May 2017.

[25] C.-L. Liu, S.-W. Fu, Y.-J. Li, J.-W. Huang, H.-M. Wang, and Y. Tsao, "Multichannel speech enhancement by raw waveform-mapping using fully convolutional networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1888–1900, 2020.

[26] E. Ceolini and S. Liu, "Combining deep neural networks and beamforming for real-time multi-channel speech enhancement using a wireless acoustic sensor network," in *IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, Oct 2019, pp. 1–6.

[27] N. Furnon, R. Serizel, I. Illina, and S. Essid, "DNN-based distributed multichannel mask estimation for speech enhancement in microphone arrays," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 4672–4676.

[28] L. Pfeifenberger, M. Zöhrer, and F. Pernkopf, "DNN-based speech mask estimation for eigenvector beamforming," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 66–70.

[29] R. Serizel, M. Moonen, B. Van Dijk, and J. Wouters, "Low-rank Approximation Based Multichannel Wiener Filter Algorithms for Noise Reduction with Application in Cochlear Implants," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 785–799, 2014.

[30] A. Bertrand and M. Moonen, "Distributed adaptive node-specific signal estimation in fully connected sensor networks — Part II: Simultaneous and asynchronous node updating," pp. 5292–5306, Oct 2010.

[31] A. Hassani, A. Bertrand, and M. Moonen, "GEVD-based low-rank approximation for distributed adaptive node-specific signal estimation in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 64, no. 10, pp. 2557–2572, 2015.

[32] M. Weintraub, "A theory and computational model of auditory monaural sound separation (stream, speech enhancement, selective attention, pitch perception, noise cancellation)," Ph.D. dissertation, Stanford, CA, USA, 1985.

[33] Z. Jin and D. Wang, "A supervised learning approach to monaural segregation of reverberant speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 4, pp. 625–638, 2009.

[34] R. Li, X. Sun, T. Li, and F. Zhao, "A multi-objective learning speech enhancement algorithm based on IRM post-processing with joint estimation of SCNN and TCNN," *Digital Signal Processing*, p. 102731, 2020.

[35] R. Gu, S. Zhang, L. Chen, Y. Xu, M. Yu, D. Su, Y. Zou, and D. Yu, "Enhancing end-to-end multi-channel speech separation via spatial feature learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7319–7323.

[36] M. Kolbæk, Z.-H. Tan, and J. Jensen, "Speech intelligibility potential of general and specialized deep neural network based speech enhancement systems," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 153–167, 2016.

[37] R. Scheibler, E. Bezzam, and I. Dokmanic, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr 2018.

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

[39] F. Font, G. Roma, and X. Serra, "Freesound technical demo," *ACM International Conference on Multimedia (MM'13)*, pp. 411–412, 2013.

[40] R. M. Corey, M. D. Skarha, and A. C. Singer, "Massive distributed microphone array dataset," 2019. [Online]. Available: https://doi.org/10.13012/B2IDB-6216881_V1

[41] G. Hinton, N. Srivastava, and K. Swersky, "COURSERA: Neural networks for machine learning – lecture 6a," 2012. [Online]. Available: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

[42] A. Narayanan and D. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013.

[43] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[44] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *2010 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2010, pp. 4214–4217.

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

[46] L. Drude, J. Heitkaemper, C. Boeddeker, and R. Haeb-Umbach, "SMS-WSJ: Database, performance measures, and baseline recipe for multi-channel source separation and recognition," *arXiv preprint arXiv:1910.13934*, 2019.

[47] S. Markovich-Golan, A. Bertrand, M. Moonen, and S. Gannot, "Optimal distributed minimum-variance beamforming approaches for speech enhancement in wireless acoustic sensor networks," *Signal Processing*, vol. 107, pp. 4–20, 2015.

[48] Y. Luo, C. Han, N. Mesgarani, E. Ceolini, and S.-C. Liu, "FaSNet: Low-latency adaptive beamforming for multi-microphone audio processing," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 260–267.

[49] M. Pariente, S. Cornell, J. Cosentino, S. Sivasankaran, E. Tzinis, J. Heitkaemper, M. Olvera, F.-R. Stöter, M. Hu, J. M. Martín-Doñas *et al.*, "Asteroid: the pytorch-based audio source separation toolkit for researchers," *arXiv preprint arXiv:2005.04132*, 2020.

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

[51] S. Doclo, M. Moonen, T. Van den Bogaert, and J. Wouters, "Reduced-bandwidth and distributed MWF-based noise reduction algorithms for binaural hearing aids," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 1, pp. 38–51, 2009.

[52] M. Souden, S. Araki, K. Kinoshita, T. Nakatani, and H. Sawada, "A multichannel MMSE-based framework for speech source separation and noise reduction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 9, pp. 1913–1928, 2013.

[53] X. Li, L. Girin, S. Gannot, and R. Horaud, "Non-stationary noise power spectral density estimation based on regional statistics," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 181–185.

[54] A. Bertrand and M. Moonen, "Topology-independent distributed adaptive node-specific signal estimation in wireless sensor networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 1, pp. 130–144, 2017.