Overlapped speech detection and speaker counting using distant microphone arrays

Samuele Cornell, Maurizio Omologo, Stefano Squartini, Emmanuel Vincent

Overlapped Speech Detection and Speaker Counting using Distant Microphone Arrays

Samuele Cornell[a,*], Maurizio Omologo[b,c,**], Stefano Squartini[a], Emmanuel Vincent[d]

[a] *Department of Information Engineering, Università Politecnica delle Marche, Italy*
[b] *Fondazione Bruno Kessler, Trento, Italy*
[c] *Alexa Speech, Amazon*
[d] *Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France*

Abstract

We study the problem of detecting and counting simultaneous, overlapping speakers in a multichannel, distant-microphone scenario. Focusing on a supervised learning approach, we treat Voice Activity Detection (VAD), Overlapped Speech Detection (OSD), joint VAD and OSD (VAD+OSD) and speaker counting in a unified way, as instances of a general Overlapped Speech Detection and Counting (OSDC) multi-class supervised learning problem. We consider a Temporal Convolutional Network (TCN) and a Transformer based architecture for this task, and compare them with previously proposed state-of-the art methods based on Recurrent Neural Networks (RNN) or hybrid Convolutional-Recurrent Neural Networks (CRNN). In addition, we propose ways of exploiting multichannel input by means of early or late fusion of single-channel features with spatial features extracted from one or more microphone pairs. We conduct an extensive experimental evaluation on the AMI and CHiME-6 datasets and on a purposely made multichannel synthetic dataset. We show that the Transformer-based architecture performs best among all architectures and that neural network based spatial localization features outperform signal-based spatial features and significantly improve performance compared to single-channel features only. Finally, we find that training with a speaker counting objective improves OSD compared to training with a VAD+OSD objective.

Keywords: voice activity detection, overlapped speech detection, speaker counting, distant microphones, spatial features

*Corresponding author
**The contribution to this work was provided by M. Omologo while he was with FBK.
Email addresses: s.cornell@pm.univpm.it (Samuele Cornell), omologo@fbk.eu (Maurizio Omologo), s.squartini@univpm.it (Stefano Squartini), emmanuel.vincent@inria.fr (Emmanuel Vincent)

1. Introduction

1.1. Motivation

In spontaneous human conversations different speakers tend to overlap with each other and, in meeting scenarios with more than two participants, the amount of overlapped speech can account for a significant portion of the total speech time, usually between 10% and 20% (McCowan et al., 2005; Watanabe et al., 2020). This phenomenon is one of the main obstacles towards fully reliable multi-party speech diarization (Ryant et al., 2018; García-Perera et al., 2020) and recognition (Watanabe et al., 2017; Vincent et al., 2018; Haeb-Umbach et al., 2019). In fact, most current techniques for speech recognition and diarization are not designed to deal directly with overlapped speech. As a result, their performance can degrade significantly in such conditions.

For this reason, Overlapped Speech Detection (OSD) is crucial to prevent back-end task performance degradation. This can be accomplished by including a reliable OSD algorithm together with Voice Activity Detection (VAD) in the very front-end part of the pipeline, possibly followed by speech separation (García-Perera et al., 2020; Watanabe et al., 2020). Speaker counting (Stöter et al., 2019) is a closely related task, which can be seen as an extension of VAD+OSD. Instead of merely identifying when there is speech and overlapped speech, speaker counting aims to directly estimate the actual number of concurrent speakers. This task, which is directly related to the problem of counting the number of coherent sources is of crucial importance for modern spoken language understanding applications, such as voice-based virtual assistants. In fact, in many real far-field speech recognition applications, encountering three or more simultaneous sources is quite common, for example in an home environment, where sound could also come from electronic devices. Speaker counting could be also employed to further help back-end tasks such as speech separation Stöter (2019) and also diarization, by extending previously proposed methods which employed only OSD information, such as Bullock et al. (2020). The accuracy of OSD, VAD and speaker counting algorithms is critical, as errors can propagate to the subsequent processing blocks, severely impacting, for example, speech recognition performance when speech segments are missed (Tong et al., 2014).

1.2. Related works

The research towards reliable OSD spans more than one decade, with the first systems relying on handcrafted features and classical machine-learning approaches. Most of these early studies focused on Gaussian Mixture Model (GMM) or Hidden Markov Model (HMM) based classifiers (Boakye et al., 2011; Vipperla et al., 2012; Charlet et al., 2013; Yella and Bourlard, 2014; Lee et al., 2016) with the exception of Geiger et al. (2013) who showed a Long-Short Term Memory (LSTM) neural network to outperform a GMM-HMM system. Boakye et al. (2011), Vipperla et al. (2012), and Yella and Bourlard (2014) reported a substantial reduction of the Diarization Error Rate (DER) on the AMI meeting corpus (Carletta et al., 2005) by removing overlapped speech segments from the segment clustering phase and performing overlap attribution afterwards.

When multiple microphone channels are available, speaker counting can be performed by clustering interchannel features (Drude et al., 2014; Pasha et al., 2017) or explicitly localizing the speakers in space (Brutti et al., 2010; Pavlidi et al., 2012), both in the single-array and multiple-array scenarios. Single-channel speaker counting is more challenging, with early works focusing on handcrafted features such as the modulation index (Arai, 2003), the mean and variance of the 7th Mel filter (Ouamour et al., 2008) or the cosine similarity between Mel Frequency Cepstrum Coefficient (MFCC) feature vectors along with pitch (Xu et al., 2013). More recently, Andrei et al. (2015) estimated the number of speakers by computing the distance between the mixture and a reference single-speaker utterance in the magnitude spectral domain.

CountNet (Stöter et al., 2019) marked a significant departure from these previous works by showing that a neural network can be trained to perform speaker counting without relying on handcrafted features, and it can even outperform humans. Andrei et al. (2019) also showed that a neural network based speaker counting algorithm can defeat human ability especially when more than three speakers are active. Kanda et al. (2020) took a different direction: they trained a neural network to perform joint speaker counting, speech recognition and speaker identification in a fully end-to-end fashion. In all these works, synthetic mixtures are employed for both training and testing and, crucially, the datasets are designed with balanced proportions of single-speaker speech, two-speaker overlapped speech, three-speaker overlapped speech, and so on. This does not match the characteristics of real-world datasets where single-speaker speech is more frequent than two-speaker overlapped speech, which is itself much more frequent than three-speaker overlapped speech. Also related to speaker counting, End-to-End Neural Diarization (EEND) firstly proposed in Fujita et al. (2019a) and subsequently improved by Fujita et al. (2019b) by using a Transformer-based architecture. In such works a neural network is trained to perform diarization by using a permutation invariant multi-label training scheme. As these systems are trained to perform diarization they can be easily adapted to perform speaker counting. In a similar manner also the recently proposed Target-Speaker VAD (TS-VAD) framework of Medennikov et al. (2020) can be employed for the purpose of counting speakers. This technique has been shown to be particularly effective even in challenging scenarios such as CHiME-6 (Watanabe et al., 2020). TS-VAD employs a neural network to estimate each speaker speech activity. However, unlike EEND and CountNet, it requires, as an additional input to the network, i-vectors (Dehak et al., 2010) for each speaker. This additional requirement makes TS-VAD more cumbersome to implement, especially for streaming applications, than both EEND and CountNet which employ end-to-end neural approaches.

Regarding OSD, Andrei et al. (2017) and Sajjan et al. (2018) recently showed that deep neural networks significantly outperform classical machine-learning approaches for this task too. Notably, Sajjan et al. (2018) evaluated four network architectures for joint VAD and OSD (VAD+OSD): a feedforward network, a 2-D convolutional network, a recurrent LSTM network and a hybrid 2-D convolutional-LSTM network. They showed that these approaches surpass

a baseline GMM-based method on both synthetic data and AMI distant-speech data, that the LSTM-based approach performs best, and that it significantly improves diarization results. More recently Kunešová et al. (2019) and Bullock et al. (2020) reported impressive OSD performance in near-field conditions, with Bullock et al. (2020) reporting up to 20% relative Diarization Error Rate (DER) reduction on the AMI headset mix. In another vein, Málek and Žďánskỳ (2020) addressed VAD+OSD by employing simple classifiers on top of pre-trained x-vector speaker embeddings (Snyder et al., 2018) and evaluated them on synthetic data corrupted by noise and artificial reverberation.

1.3. Our contribution

In this article, we unify VAD, OSD, joint VAD+OSD, and speaker counting as instances of a general Overlapped Speech Detection and Counting (OSDC) supervised classification task. We study two TCN (Bai et al., 2018) and Transformer (Vaswani et al., 2017) based architectures for this task, compare them with the LSTM-based architecture of Sajjan et al. (2018) and CountNet, and present an in-depth study of their computational efficiency. In addition, we explore the use of spatial features to aid VAD+OSD and speaker counting. As mentioned above, a number of works have shown that spatial features can be used for counting (Drude et al., 2014; Pasha et al., 2017; Brutti et al., 2010; Pavlidi et al., 2012) and VAD (Vecchiotti et al., 2019b). However, to our knowledge, no study has yet been performed where spatial features are used in conjunction with deep neural networks to tackle OSD and speaker counting directly. We perform an extensive experimental evaluation using a purposely made multi-channel synthetic dataset and two real-world, multi-microphone, distant-speech datasets: AMI (McCowan et al., 2005) and CHiME-6 (Watanabe et al., 2020). This article significantly extends and improves upon our preliminary study (Cornell et al., 2020), which did not include the Transformer-based architecture, was restricted to single-channel input and a single type of single-channel features, did not analyze the results as a function of speaker distance or angle, and did not report computational efficiency.

In detail, we first evaluate the different architectures on AMI and CHiME-6 for both VAD+OSD and speaker counting, considering single-channel features only for the sake of comparison with Sajjan et al. (2018) and Cornell et al. (2020). We show that the proposed Transformer-based network, despite having the lowest computational footprint, achieves the best performance on all tasks. We then study how its real-world performance can be further improved by adding spatial features. We examine different such features, including classical interchannel features and neural network based localization features. Also suitable early fusion and late fusion schemes for combining single-channel spectral features and spatial features are compared. The synthetic dataset is used to further study and validate our findings in a controlled environment where oracle speaker locations are known. For the sake of reproducibility, the code used to perform the experiments and to generate the synthetic dataset is made

publicly available.[1]

The remainder of this paper is structured as follows. In Section 2, we explain the multi-class classification framework adopted through this work for supervised VAD+OSD and speaker counting. Section 3 presents the proposed and existing neural architectures and Section 4 introduces the spatial features we explore for this purpose. Then, in Section 5, we describe the datasets used for the experiments and, in Section 6, we report and discuss our extensive experimental evaluation, including the comparison of computational requirements and the results achieved by single-channel and multichannel systems. Finally, in Section 7, we summarize the results obtained, draw conclusions and outline possible future work directions.

2. Overlapped Speech Detection and Counting Framework

In this work, we treat supervised VAD, OSD, VAD+OSD, and speaker counting in a unified way, as special instances of a general OSDC task. This task can be formulated as a multi-class supervised sequence labeling problem, with a different number of classes for VAD, OSD, joint VAD+OSD, and speaker counting.

We consider a parametric model $\mathcal{F}(\mathbf{X};\boldsymbol{\theta})$ which takes as input a sequence of frame-level feature vectors $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_m}\}$ and outputs a sequence of class posterior probabilities. We assume that the model may perform internal subsampling, i.e., one output frame is provided every K input frames. This is because frame-level estimation is unnecessary for most speech segmentation applications and, by employing subsampling operations, the computational burden can be reduced.

In the supervised setting, we are given the ground-truth class label sequence $\mathbf{y} = \{y_1, y_2, \ldots, y_l\}$ of length $l \leq m$, and we wish to estimate the optimal model parameters $\widehat{\boldsymbol{\theta}}$ according to a certain criterion. As in this paper we focus on neural approaches, the optimal model parameters are estimated on a suitable training set composed of N pairs of input feature sequences and corresponding class label sequences $\mathbf{T} = \{(\mathbf{X_1}, \mathbf{y_1}), \ldots (\mathbf{X_N}, \mathbf{y_N})\}$ by using Stochastic Gradient Descent (SGD) to minimize the cross-entropy loss between the estimated frame-level posterior probabilities and the true class distribution.

In this framework, VAD and OSD can be treated either separately as binary classification tasks (speech vs. non-speech, overlap vs. non-overlap), or jointly as a three-class (non-speech, single speaker, overlapped speech) problem. Similarly, speaker counting can be formulated as an C-class classification task where C is equal to the maximum possible number of overlapping speakers plus one. While this approach is not the only one for supervised speaker counting, it has been found to be the most effective (Stöter et al., 2019), provided the maximum possible number of concurrent speakers is known.

[1] github.com/popcornell/OSDC

3. Neural Architectures for OSDC

We study four neural network architectures for tackling the OSDC task.

3.1. Long-Short Term Memory (LSTM)

The first one is the best neural network for joint VAD+OSD among the ones examined by Sajjan et al. (2018) which, to our knowledge and with the exception of our preliminary work (Cornell et al., 2020), achieves the best reported performance on AMI single-channel distant-speech data.

It consists of a unidirectional LSTM layer with a hidden size of 512 neurons, followed by 3 dense layers with 1024, 512 and 256 neurons, respectively. A final $256 \times N$ pointwise convolutional layer along with softmax is used to output the probability of each frame belonging to one of the N classes (e.g., $N = 3$ for VAD+OSD). This network features a total of 2 million parameters and generates one output vector for every input frame given a context of 11 frames (current frame plus 5 past and 5 future frames).

As the original architecture lacked any normalization technique, in our experiments we added batch normalization (Ioffe and Szegedy, 2015) before each dense layer activation as well as layer normalization (Ba et al., 2016) on the input features. This, coupled with data-augmentation, allows us to improve performance over the original network as it will be shown in Section 6.5.

3.2. Hybrid Convolutional-Recurrent Neural Network

We also consider the best CountNet architecture among the 5 different networks compared by Stöter et al. (2019). This network is a hybrid Convolutional-Recurrent Neural Network (CRNN) , composed of a 2-D Convolutional Neural Network (CNN) block followed by an RNN block. The main idea behind this architecture is that the CNN extracts a local representation of the input features while the RNN deals with long-term temporal modeling, thus combining the advantages of both CNNs and RNNs.

Input features of shape $F \times T$ are fed to the CNN which is composed of two blocks, each composed of two 2-D convolutional layers with kernel size 3×3 followed by ReLU activation and a 3×3 max-pooling subsampling operation. A total of 4 convolutional layers is thus employed with 64, 32, 128 and again 64 channels, respectively. Dropout (Srivastava et al., 2014) is applied on the output of the CNN and the representation is fed to an LSTM layer with a hidden size of 40. As an LSTM operates on 2-D sequences while the output of the CNN is a 3D tensor with channel, frequency, and time dimensions, a 2-D sequence is obtained by stacking the frequency dimension onto the channel dimension.

Stöter et al. (2019) performed an additional max-pooling operation on the whole time dimension in order to output a single prediction for the entire input because they aimed to count the maximum number of speakers in the whole sequence. Here, as explained in Section 2, we are interested in estimating the number of speakers in each time frame instead so we omit this final pooling layer. In this way, the network generates one output vector for every 6 feature vectors in input. As this architecture also originally lacked any normalization

strategy, we added batch normalization after each convolutional layer and layer normalization at the input.

3.3. Temporal Convolutional Network

In addition to the above two state-of-the-art architectures, we consider a TCN architecture for the OSDC task. This type of architecture has been shown to achieve state-of-the-art performance in many sequence-related tasks (Bai et al., 2018) and for source separation (Luo and Mesgarani, 2019).

TCNs rely on multiple stacked dilated convolutional layers whose dilation factor increases progressively as depth increases. This makes it possible to greatly expand the receptive field, such that upper layers can have access to long-term contextual information without any pooling operation. This in turn allows TCNs to outperform recurrent models in some tasks (Bai et al., 2018). In fact, because they are based only on convolutional operations, TCNs have several benefits with respect to RNNs. First, being feedforward, they are not affected by the vanishing gradient problem which plagues RNNs, as skip-connections and residual connections can be used to backpropagate the gradient unscathed down to the very first layers. Second, in RNNs the information about the past must be contained in the hidden state. This makes it difficult to learn very long-term dependencies as all relevant information about the past must be squeezed into this finite-sized representation. On the contrary, TCNs process the whole sequence and, because no downsampling is performed, the information at all steps is preserved in all layers. Finally, as no recurrent operations are employed, TCNs are significantly faster than recurrent models in both the training and inference phases. However, the fact that the representation is not pooled leads TCNs to have large memory requirements in general, especially if a very wide receptive field is desired.

The architecture we employ here (Cornell et al., 2020) is depicted in Fig. 1. It is inspired from MobileNet (Howard et al., 2017) and Conv-TasNet (Luo and Mesgarani, 2019). Input frame-level feature vectors of size F (e.g., log-Mel filterbanks) are fed to a layer normalization (Ba et al., 2016) layer followed by an $F \times 64$ 1D pointwise convolutional layer (denoted as *conv 1x1*) and by $R = 3$ blocks of $X = 5$ residual blocks (*res blocks*) with 1D dilated convolutions, where the dilation factor increases in each block as $2^0, 2^1, \ldots, 2^{X-1}$. Each residual block consists of a 64×128 pointwise convolutional layer followed by batch normalization and activation, a dilated depthwise separable 128×128 convolutional layer (*d-conv*) followed by batch normalization and activation, and another 128×64 pointwise convolution which squeezes the representation back so that it can be summed with the input. We use PReLU (He et al., 2015) as the activation function in all residual blocks and a kernel size of 3 in depthwise dilated convolutions.

3.4. Transformer

Finally, we propose a Transformer-based architecture for OSDC. Transformers, which were originally proposed by Vaswani et al. (2017) for natural language

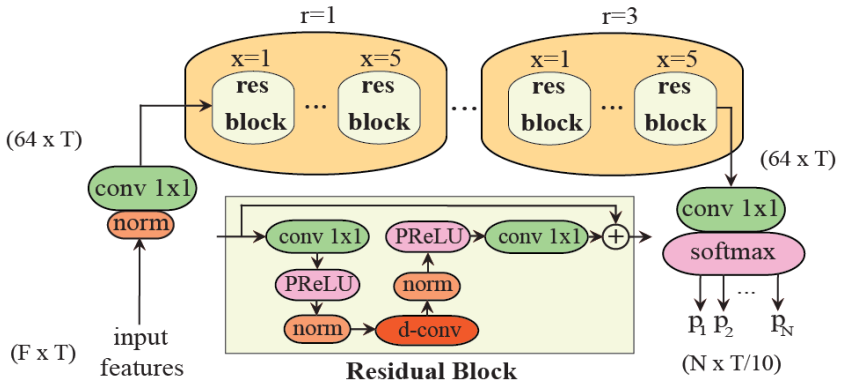

Figure 1: Proposed TCN architecture for the OSDC task.

processing applications, are pure attention-based models which have been shown recently to achieve state-of-the-art performance in many speech processing tasks including diarization (Fujita et al., 2019b). They have several advantages over recurrent models, including faster inference speed and better modeling of long-term dependencies. Being feedforward models, the whole sequence is attended at once, eliminating any recurrence and any need for an internal hidden state to keep track of past elements. For this reason, they exhibit the same advantages as TCNs over RNNs, even if their inherent functioning is significantly different. Similarly to TCNs and while being much faster than RNNs, Transformers also have higher memory requirements, due to the fact that the attention mechanism grows as $\mathcal{O}(n^2)$ in memory with n the length of the input sequence.

Our Transformer-based architecture is depicted in Figure 2 and, as it can be seen, has some input and output blocks in common with the previously described TCN network. To counter the quadratical memory growth induced by the attention mechanism, we adopt a concatenate-subsample (*cat-pool*) operation over the input feature vectors. For each frame, we concatenate the feature vectors from C past frames and C future frames with the current one. Afterwards, we subsample this representation on the frame axis by a factor of S. In this way, the information contained in the temporal dimension is effectively transferred to the feature dimension with a resampling factor of C/S the original rate. This concatenated and pooled representation is then fed to a layer normalization layer followed by a pointwise convolutional layer (*conv 1x1*) which shrinks the representation to a predefined size H to reduce the memory requirements of subsequent blocks, allowing us to process longer sequences or, alternatively, to reduce the computational footprint of the model as it will be shown in Section 6.4. Sinusoidal positional encoding is added right after this bottleneck convolutional layer and the result is fed to a succession of R Transformer Encoder blocks, each composed of two residual sub-blocks.

The structure of each Transformer Encoder block is identical to the one proposed by Vaswani et al. (2017) with the exception that, in our architecture, layer

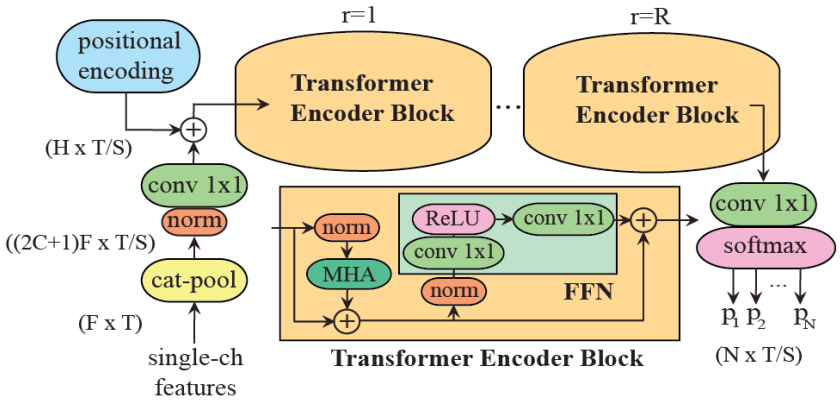

Figure 2: Proposed Transformer architecture for the OSDC task.

normalization is performed at the beginning of each residual block rather than in the end. Indeed, Nguyen and Salazar (2019) recently found that this results in better performance as well as faster convergence. The first residual block consists of a normalization layer followed by a Multi-Head Attention (MHA) layer and dropout. The second one consists of a normalization layer followed by a position-wise feedforward neural network (FFN) composed of one dense layer,[2] a ReLU activation followed by dropout, and another dense layer which projects the hidden representation back. As in the TCN model, a final $H \times N$ pointwise convolutional layer followed by softmax is used at the output.

4. Spatial Features and Feature Fusion Schemes for OSDC

Intuitively, spatial features can help VAD, OSD and speaker counting. For example, OSD and speaker counting can benefit from knowing whether the sound comes from one or more Directions of Arrival (DoAs). VAD can also benefit from spatial features to distinguish speech, which is usually directional, from noise, which can be spatially diffuse.

In fact, as mentioned in Section 1.2, many works have tackled speaker counting by framing it as a localization problem. These works resort to DoA estimation methods based on generalized cross-correlation with phase transform (GCC-PHAT) (Knapp and Carter, 1976) as in (Brutti et al., 2010; Drude et al., 2014), magnitude-squared coherence (MSC) (Pasha et al., 2017) or simple cross-power spectrum (Pavlidi et al., 2012; Walter et al., 2015). The speaker number is estimated via a direct approach such as in (Brutti et al., 2010) by counting peaks in GCC-PHAT based acoustic maps or by clustering methods, where speaker clusters are identified by iterative grouping of complex-valued time-frequency coefficients (Drude et al., 2014), magnitude squared coherence feature vectors

[2] Note that dense layers are equivalent to *conv 1x1*.

(Pasha et al., 2017), or DoAs estimated over single-source time-frequency zones (Pavlidi et al., 2012) or individual time-frequency bins (Walter et al., 2015).

Recently, a series of works have proven that neural network based localization is more robust than signal-based methods in reverberant and noisy environments. In these works, a neural network is trained to estimate the DoA on a synthetic dataset for which the true position of the sources is known. Input features include GCC-PHAT (Xiao et al., 2015), cosine-sine interchannel phase difference (CSIPD) features (Sivasankaran et al., 2020), the phase spectra of all channels (phasemap) (Chakrabarty and Habets, 2017), the magnitude and phase spectra (Adavanne et al., 2018), or the raw waveform (Vecchiotti et al., 2019a).

4.1. Signal-based Spatial Features

In this paper, for what concerns signal-based spatial features, we explore the interchannel phase difference (IPD) and CSIPD, as they have been shown in the aforementioned works to work well in reverberant and noisy environments. In particular, our choice of IPD instead of phasemap is justified by the fact that, both in AMI and CHiME-6, microphones are close to each other and thus some microphone pairs can be discarded as they do not add much spatial diversity at 16 kHz. On AMI, we consider only those pairs of microphones with maximal distance from each other, i.e., the 4 pairs formed by opposite microphones in each circular array instead of all 28 possible pairs. On CHiME-6, due to the asymmetrical placement of microphones in Kinect devices, we consider the 3 pairs formed by channels 1 and 4, channels 2 and 4, and channels 3 and 4. The IPD or CSIPD features of all pairs are then concatenated together over the frequency dimension. Thus, in these contexts, using interchannel features allows us to reduce the feature size with respect to the phasemap and hence save computational resources with practically no loss in spatial information.

IPD and CSIPD features are tightly related and derive from the phase spectrum. Denoting by $x_i(n, f)$ and $x_j(n, f)$ the STFT of the i-th and j-th microphone signals, where n and f are respectively the frame and frequency index, the IPD $\phi_{i,j}(n, f)$ between channel i and j is given by

$$\phi_{i,j}(n, f) = \angle x_i(n, f) - \angle x_j(n, f), \tag{1}$$

where $\angle(.)$ is the function returning the phase from the input complex value. The IPD feature vector in time frame n is then defined as

$$\mathbf{IPD}(n) = \left[\phi_{i,j}(n, 0), \phi_{i,j}(n, 1), \ldots, \phi_{i,j}\left(n, F/2\right)\right]^T, \tag{2}$$

with F the FFT size. The CSIPD feature vector in time frame n can be obtained directly from the IPD feature vector and is another way of encoding the information contained in it by using its cosine and sine values:

$$\mathbf{CSIPD}(n) = \left[\cos\phi_{i,j}(n, 0), \sin\phi_{i,j}(n, 0), \ldots, \sin\phi_{i,j}\left(n, F/2\right)\right]^T. \tag{3}$$

An important property of CSIPD is that the GCC-PHAT angular spectrum for a given microphone pair (or the SRP-PHAT spectrum when there are 3 or

more microphones and all pairs are considered) can be expressed as a linear transformation of the CSIPD feature vector (Sivasankaran, 2020). When these features are to be input to a neural network model, there is therefore no benefit in using the GCC-PHAT or SRP-PHAT angular spectra as inputs instead, since this linear transformation can be learned by the neural network itself. This was confirmed by our experiments, so we do not report results obtained with GCC-PHAT or SRP-PHAT features in the following.

4.2. Neural Network-based Localization Features

As an alternative, we also consider the strategy of training a neural network to estimate the DoAs of multiple overlapped speakers on a suitable synthetic dataset for which the true DoAs are known. The embeddings extracted by some intermediate layer of this network can then be used as "higher-level", possibly more robust spatial features to be employed in the OSDC system. In this work, we adopt the multi-speaker localization method of Chakrabarty and Habets (2017), where the space of DoAs is discretized and the neural network is trained to estimate the posterior probability that a speaker is active for each discrete DoA by minimizing the sum of binary cross-entropies across all discrete DoAs. Binary cross-entropy is used as the cost function since multiple concurrent speakers with different DoAs can be active at the same time.

In detail, even for localization, we use the networks outlined in Section 3 by modifying the output layer which is replaced with mean pooling over the sequence dimension and a new linear layer with output size D followed by sigmoid activation, where D is the number of discrete DoAs considered. The network representation before the mean pooling operation is then employed as a spatial feature vector for OSDC systems.

One advantage of neural network-based features over signal-based features is that joint fine-tuning of the two networks can be performed, thus optimizing the localization feature extraction network for OSDC applications. However, it must be noted that the computational footprint significantly increases by using neural network based features. Also, the fact that true source DoAs are needed for training necessitates the use of a synthetic training dataset, which can be mismatched with real-world data.

4.3. Fusion schemes

Spatial features are not sufficient for reliable OSDC when used alone. For example, directional noise sources may sometimes be confused with speech, or concurrent speakers can have the same DoA. They must hence be combined with single-channel spectral features, such as log-Mel spectra. We consider two different fusion schemes for this combination, which we call early and late fusion.

These fusion schemes are illustrated in Figure 3 for the Transformer-based network. In early fusion, the two features are stacked together in the very first layer of the neural network. Layer normalization on spatial features is performed separately prior to concatenation. In late fusion, after layer normalization, the spatial features are injected before each Transformer Encoder Block (*TE Block*),

using Feature-wise Linear Modulation FiLM (Perez et al., 2018). In this way, each block of the architecture can focus on a different aspect of the input spatial features since they are available even in deeper layers. As the spatial and single-channel features are concatenated together in early fusion, they must have same temporal length. Thus, for proposed Transformer network we employ the same cat-pool operation also on spatial features prior to concatenation. The same argument applies also for late fusion where instead spatial features are used to modulate activations at multiple layers.

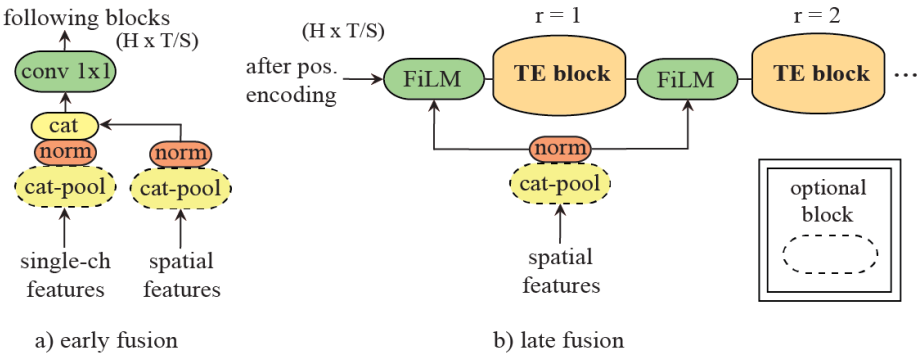

Figure 3: Fusion strategies for single-channel features and spatial features for the proposed Transformer architecture: a) early fusion, b) late fusion. TE stands for Transformer Encoder.

5. Datasets

We conduct experiments on two real-world multi-microphone datasets: AMI and CHiME-6. Moreover, we also use a synthetic dataset to further study, in a controlled situation, the use of spatial features to improve the performance of OSDC systems.

5.1. Synthetic Dataset

We simulate multi-speaker mixtures captured by a single microphone array. Clean speech utterances are taken from Librispeech (Panayotov et al., 2015) *train-clean-100* for training, *dev-clean* for validation, and *test-clean* for test. The Montreal Forced Aligner (MFA) (McAuliffe et al., 2017) is used to split these original Librispeech utterances in order to obtain shorter "sub-utterances" for each speaker. This splitting is performed whenever pauses of more than 150 ms are encountered. MFA is also used, in parallel, to obtain ground truth word-level speaker activity. For each mixture, we sample from 1 to 4 different speakers, and, for each speaker one clean speech sub-utterance is sampled. The starting time of each speaker sub-utterance is sampled independently from an exponential distribution. In this way, by varying the decay rate parameter, the amount of overlap between the speakers and the amount of silence can be controlled. A different acoustic scenario is sampled for each mixture. We

simulate a rectangular room whose size is varied between 10 and 60 m^2. The position of each speaker is chosen randomly inside the room but with some constraints. Namely, the speakers cannot be less than 0.5 m from each other and from the walls. We consider a 4-microphone linear array placed randomly with respect to the walls, whose height with respect to the floor can vary between 1.7 and 2 m and whose distance to the closest wall is larger than a minimal distance which is varied between 10 and 30 cm. We use the gpuRIR (Diaz-Guerra et al., 2018) toolkit for room simulation with a T60 reverberation time uniformly sampled between 0.2 and 0.6 s. Anechoic noise from Furnon et al. (2020) is also employed to make the dataset more realistic. The positions of noise sources inside the room are selected with the same criteria as the speakers' ones. The whole synthetic dataset consists of a total of 10 k mixtures ($\sim$ 23 hours) for training, 2 k for validation and 2 k for test ($\sim$ 4.6 hours).

5.2. AMI

The AMI Corpus (McCowan et al., 2005) is over 100 h of meeting recordings. Each meeting has been recorded by a variety of devices including cameras, microphone arrays, and per-speaker headset and lapel microphones and has from 3 to 5 participants. Ground truth speaker activity was obtained by human annotators from close-talk speaker-worn microphones while distant speech was recorded by two 8-microphone circular arrays, each with a 10 cm diameter: one placed at the end and another at the centre of the meeting table used by the participants.

5.3. CHiME-6

The CHiME-6 corpus comprises dinner party recordings. The recordings are divided into 20 sessions for a total of more than 60 h of data. In each session, 4 speakers are recorded in a real home environment consisting of different rooms. Due to the particular setting, it features conversational speech and low Signal-to-Noise Ratio (SNR). Recordings from binaural microphones worn by each speaker are provided along with distant speech captured by 6 array devices with 4 microphones each for a total of 24 microphones. Two different annotations are provided for the start and end time of every utterance: looser ones geared towards Automatic Speech Recognition (ASR) and tighter ones obtained via forced-alignment. The latter ones are more suitable for evaluating VAD and diarization systems and we use them in the following.

6. Experimental Results

In the following, we evaluate the neural architectures in Section 3 and the spatial features and feature fusion schemes in Section 4 on the datasets described in Section 5. Firstly, in Section 6.1, we define and motivate the chosen performance metric. In Section 6.2, we outline the training and testing procedure adopted in our experiments and, in Section 6.3, we highlight the impact of different choices of hyperparameters and single-channel input features for the

Transformer-based architecture. Then, in Section 6.4, we provide an analysis of the computational footprint of the four considered neural architectures when applied to single-channel data and, in Section 6.5, we report their OSDC performance on AMI and CHiME-6. Finally, in Section 6.6, we assess the impact of spatial features on the best single-channel system: we explore different spatial features, fusion schemes and number of microphone pairs, and evaluate the results on AMI, CHiME-6 and the proposed synthetic dataset.

6.1. Evaluation Metric

On real-world data, VAD, OSD and speaker counting tasks are affected by class imbalance. This imbalance, which arises from intrinsic characteristics of human conversations, can be more or less severe depending on the context. This can be seen in Table 1, which reports the class statistics on AMI and CHiME-6 for the counting task.[3] Due to its informal, "cocktail-party" scenario, the CHiME-6 dataset exhibits a slightly higher proportion of overlapped speech than the AMI dataset, which consists of meetings. Nevertheless, in both datasets, the proportion of 4-speaker and 3-speaker overlap is very small. The imbalance is less severe for VAD and OSD tasks but, even for these, the choice of the evaluation metric can be crucial.

Table 1: Frame-level class frequency (%) for the speaker counting task on the AMI and CHiME-6 development and evaluation sets.

Class frequency		0-spk	1-spk	2-spk	3-spk	4-spk
AMI	dev	15.87	67.17	13.95	2.59	0.42
	eval	15.12	68.39	12.63	3.09	0.76
CHiME-6	dev	24.05	54.25	17.74	3.49	0.47
	eval	33.47	51.52	12.03	2.46	0.51

We argue that metrics such as accuracy and precision-recall, as used respectively by Sajjan et al. (2018) and by Kunešová et al. (2019) and Bullock et al. (2020), do not provide a fair evaluation of OSDC algorithms on real-world data due to this fundamental imbalance. For example, concerning OSD on the AMI evaluation set, an accuracy of 83.7% can be reached by labeling all the material as no-overlap. As it has been observed by Cornell et al. (2020, Table 5), this leads to small accuracy differences even for classifiers with drastically different performance. In this scenario, precision and recall are a better choice than accuracy. However, similarly to accuracy, their value depends on the choice of the detection threshold which can be application-specific (e.g., a different threshold for diarization and speech recognition is often desirable). This does not allow for a fair comparison between different OSDC algorithms.

[3] We disregard the 5-speaker overlap class on AMI since it does not occur in practice.

For these reasons, we propose the use of Average Precision (AP) metric which summarizes the precision-recall curve and is widely used, for example, in object segmentation (Lin et al., 2014), information retrieval (Kishida, 2005) and other tasks exhibiting strong class imbalance. It can be obtained from precision P and recall R at the k-th threshold as:

$$AP = \sum_{k=1}^{M} (R_k - R_{k-1}) P_k, \tag{4}$$

where M is the total number of unique thresholds considered. The number of elements in this set is upper bounded by the number of unique elements in the classifier output probabilities vector. In this work we use each time the maximum number of possible thresholds to compute AP. As it can be seen, AP has the advantage that it does not depend on a particular threshold, making it both more robust to imbalanced data and more suitable for comparison purposes. In all experiments, AP scores are computed on 10 ms time frames.[4]

AP in Equation (4) is suitable only for binary classification tasks such as VAD and OSD. However, it can be extended easily to multi-class classification problems such as Speaker Counting with a leave-one-out classification strategy: e.g. AP for the 1-spk class can be obtained by considering an equivalent binary task where the true positives are the frames correctly classified as 1-speaker and the true negatives are the frames classified as silence (0-spk), 2-speaker (2-spk), 3-speaker (3-spk), or 4-speaker (4-spk). In a similar way one can compute VAD and OSD AP from a neural network trained to perform Speaker Counting or VAD+OSD. For example, for a Speaker counting algorithm, VAD predictions can be obtained by summing the probabilities for the classes with at least one speaker: 1-spk, 2-spk, 3-spk, and 4-spk, thus obtaining the total probability of speech; OSD by summing classes with at least 2 speakers: 2-spk, 3-spk, and 4-spk.

Unless stated otherwise, in each Table, we highlight in bold font the best result and the ones which are statistically equivalent to it (if any) with $p = 0.001$. Because we found the distribution of the AP metric to be highly non-gaussian, we use the Wilcoxon-Signed Rank non-parametric test (Demšar, 2006).

6.2. Training and Testing Procedure

In the following experiments, we use the exact same training and testing procedure as in our preliminary work (Cornell et al., 2020). This allows the results to be directly comparable. In detail, we train all models using RAdam (Liu et al., 2020) on 5 s chunks obtained from training signals with 50% overlap. The last chunk is discarded if shorter. Hyperparameters such as batch size, learning rate and dropout rate are tuned for each network, dataset and training objective (speaker counting or VAD+OSD) on the development set. Inference

[4] The sequence output by the Transformer model is stretched by a factor of S, in order for the number of input and output frames to be equal, similarly to the other models.

is performed by using a sliding window approach is used where the logits of overlapping blocks are averaged to obtain the final estimate. Popular speech processing toolkits such as Pyannote (Bredin et al., 2020) use this approach. In this work we use a sliding window of 3 s with 50% overlap.

In our preliminary work (Cornell et al., 2020), we found that using training targets obtained via Forced-Alignment (FA) brings considerable improvement even when manual annotation is used as the ground truth in the testing phase. We also studied the efficacy of FA as an automatic labeling procedure for speech segmentation applications using synthetic data and we found that, when close-talk worn microphones are employed, it can be considered reliable even in overlapped speech regions and challenging SNR conditions. Thus, we employ FA labels to train OSDC models on both AMI and CHiME-6. In detail, we use the Kaldi (Povey et al., 2011) recipes for AMI and CHiME-6 and get the segmentation from the *tri3* GMM-HMM speech recognition model.

The results on the test set are evaluated using the official annotation, which is manual in the case of AMI and FA-based in the case of CHiME-6. In fact, the FA-based annotation of the CHiME-6 development and evaluation sets was obtained with similar FA procedure as used here.

Moreover, to further improve performance on real-world data and counteract class imbalance, we resort, in our experiments, to the data-augmentation strategy described by Cornell et al. (2020), where it was shown to bring significant improvements. This data-augmentation technique, which is itself an extension of the one proposed by (Bullock et al., 2020), consists of on-the-fly creation, at training time, of new concurrent speaker examples by overlapping 2, 3, and 4 random single-speaker chunks from the original dataset in order to re-balance the classes. To further increase the training material, a random gain factor sampled from $\mathcal{N}(\mu = -16.7, \sigma = 4)$ in dB scale is applied to each chunk independently. In this way, we augmented the original AMI data by a factor of 70% and CHiME-6 data by 40 %. This augmentation factor is tuned for each dataset using the development set. In parallel, to improve generalization, we also use SpecAugment (Park et al., 2019) on both single-channel and spatial features separately.

6.3. Choice of Transformer Hyperparameters and Single-Channel Features

In Table 2, we show the hyperparameter space explored for the proposed Transformer-based architecture. We varied number of future and past frames (C) and subsampling factor (S) used in cat-pool operation as well as size of hidden representation (H), number of attention heads, size of feed-forward neural network hidden layer (FFN size) and number of transformer encoder blocks (R). The hyperparameters were tuned on the development set of AMI, for fair comparison with Sajjan et al. (2018) who also optimized his LSTM model on AMI. The models were trained to perform VAD+OSD according to the framework introduced in Section 2. The best combination was selected using two criteria: overall VAD+OSD performance and inference-time computational footprint, to give an overview of how much demanding the model is when used in practical

applications. In fact, if the OSDC model has a modest computational burden, using it at the very first stage of a speech processing pipeline has the advantage of lowering the computational requirements of the whole pipeline, as subsequent processing can be applied only when needed. Moreover, models with modest computational requirements allow for deployment on mobile and edge-computing devices.

Table 2: Hyperparameter space explored for the Transformer-based architecture. The best combination of hyperparameters is highlighted in bold.

Hyperparameter	C	S	H	heads	FFN size	R
Values	(**7**, 5)	(**10**, 5)	(256, **384**)	(**4**, 8, 16)	(**1024**, 2048)	(2, **4**, 8)

In Table 3, we show the VAD and OSD performance on the AMI development set, as well as the total number of floating point operations (FLOP) and total memory consumption (Mem) with the best combination of hyperparameters (Best) and when changing the value of one hyperparameter at a time. FLOP and Mem are computed with a 300-frame (3 s) dummy 80-dimensional feature sequence (matching the 80 log-Mel features used in the following experiments), generated from a uniform distribution. These computational footprint figures are estimated using the built-in profiler in the Pytorch toolkit and the Performance Application Programming Interface (Terpstra et al., 2009). Several observations can be made. First, the choice of hyperparameters does not affect the VAD performance, which is arguably a simpler task than OSD and is more easily tackled by the network. Second, doubling the number of Transformer Encoder blocks only marginally improves performance at the cost of a significant increase of the computational footprint. Third, increasing time resolution by halving the sub-sampling rate also significantly increases the computational requirements without bringing significant benefits, meaning that a resolution in the order of 100 ms is enough in the application scenario considered here.

Table 3: VAD and OSD AP (%) and computational footprint of the Transformer-based architecture on the AMI development set for different architecture hyperparameter values.

Model Parameters	FLOP [10^6]	Mem [10^6]	AP	
			VAD	OSD
Best	85.6	3.3	**98.5**	57.4
S = 5	166.8	6.9	**98.5**	57.5
R = 8	161.0	6.2	**98.5**	**57.8**
heads = 8	85.4	3.6	**98.5**	56.9
FFN size = 2048	153.1	5.1	**98.5**	**57.6**

In Table 4, we report the results achieved by the proposed Transformer-based architecture on the AMI development set for different choices of single-channel input features. In the past, Sajjan et al. (2018) and Stöter et al. (2019) explored

different single-channel features for the LSTM and CountNet architectures: Sajjan et al. (2018) used gammatone filterbanks, log-Mel and other features such as kurtosis and spectral flatness, while Stöter et al. (2019) explored magnitude STFT spectra, log spectra and 40 Mel-scale filterbanks. In both studies, the features were extracted with a 25 ms window and 10 ms hop-size. Hereafter, we consider magnitude spectra computed over 32 ms and 64 ms windows (512 and 1024 samples respectively), 40 and 80 log-Mel, 40 and 80 gammatone filterbanks, and 20 and 40 MFCCs instead. All these features were computed with a 10 ms hop-size. Regarding MFCCs, we used 20 and 40 Mel bands, respectively. A window of 25 ms was used for log-Mel, gammatone and MFCCs. We can see that OSD and to a lesser extent VAD performance correlate with frequency resolution. In fact, especially for OSD, the use of compact features such as MFCCs, 40 log-Mel or 40 gammatone filterbanks leads to a loss in performance. These results partially agree with the findings of Sajjan et al. (2018), who found 64 gammatone filterbanks to be superior to 40 log-Mel features for OSD.

Table 4: VAD and OSD AP (%) achieved by the Transformer-based architecture on the AMI development set with different choices of single-channel features.

AP	MagSpec		Log-Mel		Gammatone		MFCC	
	512	1024	40	80	40	80	20	40
VAD	**98.5**	**98.5**	98.4	**98.5**	98.4	**98.5**	98.3	98.4
OSD	**61.1**	**61.0**	58.2	**61.0**	58.0	**59.8**	56.8	58.4

Because no statistical difference was found between 80 gammatones and 80 log-Mel and higher-resolution features (e.g., 64 ms magnitude spectra) did not result in higher performance, we ultimately decided to use 80 log-Mel features in the following.

6.4. Computational Footprint Comparison Across Architectures

In Figure 4 we report the total number of floating point operations (FLOP), the total memory usage and the inference time in clock cycles for the four considered network architectures as a function of the input signal duration from 1 s to 100 s. Inference time is computed over batches of 64 examples in order to get reliable estimates. As we are interested in comparing only the architectures, we use the same single-channel features for all architectures, namely 80 log-Mel features with 25 ms window and 10 ms hop-size. An Intel i9-10920X CPU is employed to perform the comparison.

As expected, regarding inference speed, the RNN-based architectures (LSTM and CRNN) are slower than the TCN and the Transformer, which do not employ recurrence. A similar trend is observable in the FLOP plot, with the difference that the CRNN has a much higher FLOP count than the other architectures due to the use of 2-D convolutions, despite the fact that it is slightly faster than the LSTM architecture as it employs pooling operations and the CNN part is parallelizable. The use of 2-D convolutions also increases the CRNN

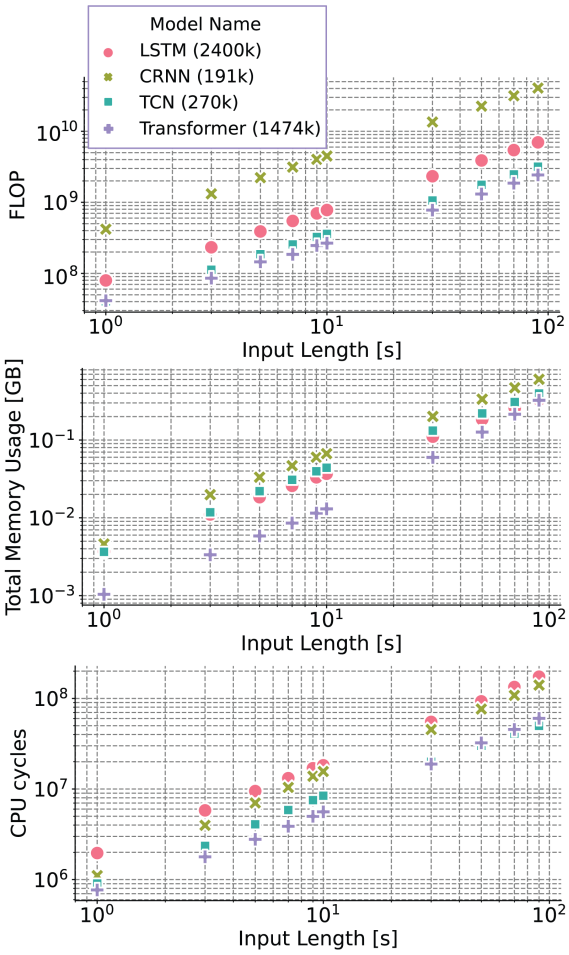

Figure 4: Inference-time computational footprint for the four considered neural network architectures as a function of the input signal duration. Top: number of floating point operations (FLOP). Middle: Total memory usage in GB. Bottom: number of CPU clock cycles. The numbers in parentheses in the legend indicate the number of model parameters. The two axes are in log-scale.

memory footprint with respect to the other architectures. The small number of parameters employed in the TCN leads to similar memory footprint as the LSTM architecture.

Overall, the proposed Transformer architecture is the most efficient according to the three criteria despite having the second largest number of parameters after the LSTM. Due to the cat-pool operation, the total memory usage is kept contained and grows almost linearly until a duration of 100 s. In practice, due to the fact that OSDC typically requires a context of a few seconds only, inference is never performed directly over such long signals. In fact, a sliding window

approach, as explained in Section 6.2 is employed. More generally, all architectures, including previously proposed LSTM and CRNN ones, attain overall computational resources figures which are suitable for edge devices deployment. For example, the FLOP count for one second of audio is comparable to the one reported by Choi et al. (2019) for keyword spotting in smartphone devices.

An important take from these results is also that the number of parameters, which is widely used as a gauge for model computational burden, does not correlate well with the latter and can be deceptive when comparing very different architectures.

6.5. Single-Channel Experimental Results

We now evaluate the performance achieved by the four architectures on the AMI and CHiME-6 distant speech datasets. For the sake of comparison with Sajjan et al. (2018) and Stöter et al. (2019), we use single-channel features only, namely 80 log-Mel features with 25 ms window and 10 ms hop-size.

Each architecture is trained and evaluated according to two different tasks: VAD+OSD and speaker counting. Indeed, we are interested in assessing the feasibility of VAD+OSD and speaker counting on real-world data. Speaker counting, as already said, has the advantage of providing more information to downstream tasks, but it is plagued by extreme class imbalance. VAD+OSD, by contrast, does not provide any clue about concurrent speakers, but exhibits a less extreme class imbalance.

Concerning AMI, to allow direct comparison with previous works (Sajjan et al., 2018; Cornell et al., 2020), data from all microphone channels is used during training while testing is performed on the first microphone of array 1. Regarding CHiME-6, training is also performed using all microphone channels from all array devices but, when evaluating, we consider for each array the first channel and then average the outputs of single-channel systems across all arrays because of the multi-room environment of CHiME-6.[5]

In Table 5, we report the VAD and OSD results obtained when training the models with a VAD+OSD objective. It can be seen that the AP figures on both datasets are considerably higher for VAD than for OSD. This is expected since OSD is inherently a more challenging task than VAD. As also expected, the performance is better on AMI than CHiME-6, as CHiME-6 is arguably a much more challenging dataset, having lower SNR due to the more unconstrained setting. The proposed Transformer architecture performs on-par or better than the other architectures, with the TCN architecture closely following. LSTM and CRNN perform significantly worse, despite the addition of normalization layers which were not present in the respective original works of Sajjan et al. (2018) and Stöter et al. (2019).[6]

[5] The single-channel evaluation protocol for CHiME-6 differs from the multichannel protocol adopted by Cornell et al. (2020), who averaged the outputs of single-channel systems across all 24 microphones instead.

[6] These normalization layers do improve performance, as can be seen by comparison with the results reported in our preliminary work (Cornell et al., 2020) which did not include them.

Table 5: VAD and OSD AP (%) achieved by the four considered neural network architectures on the AMI and CHiME-6 evaluation sets using single-channel features and VAD+OSD as a training objective.

VAD+OSD Model	VAD		OSD	
	AMI	CHiME-6	AMI	CHiME-6
LSTM	95.4	93.4	34.3	28.7
CRNN	96.7	93.8	38.9	33.2
TCN	**98.5**	**94.3**	54.2	49.0
Transformer	**98.5**	**94.3**	**57.8**	**49.9**

Similarly, Tables 6 and 7 report the speaker counting results achieved on the evaluation sets of AMI and CHiME-6, respectively, when training the models with a counting objective. The fact that the AP for the 0-spk class is remarkably lower on AMI is a rather unexpected result, as it features a much higher SNR than CHiME-6 overall. This could be explained by class imbalance since, as reported in Table 1, the proportion of 0-spk in AMI is significantly lower than in CHiME-6. The proposed Transformer architecture achieves the best figures overall on both datasets. In general, compared to the 0-spk and 1-spk classes, the AP degrades considerably for the 2-spk class and even more so for the 3-spk and 4-spk classes. This suggests that the data-augmentation strategy, is only able to partially compensate for the extreme imbalance of 3-spk and 4-spk classes. Therefore, it can be said that speaker counting is still far from being reliable on real-world data.

Table 6: Speaker counting AP (%) achieved by the four considered neural network architectures on the AMI evaluation set using single-channel features and counting as a training objective.

Counting Model	0-spk	1-spk	2-spk	3-spk	4-spk
LSTM	47.0	82.4	24.7	6.4	**0.02**
CRNN	49.8	84.2	34.8	9.2	**0.03**
TCN	**50.7**	86.1	40.4	11.3	**0.03**
Transformer	**50.9**	**87.2**	**41.8**	**11.2**	**0.03**

In Table 8 we compare the performance of Transformer models trained to perform either VAD, OSD, VAD+OSD or counting for the VAD and OSD tasks. For each dataset, we report the evaluation set performance and, in parentheses, the development set performance. Regarding VAD, the choice of the training objective has little impact on performance on all datasets. Regarding OSD, interestingly, the model trained to perform speaker counting, which is inherently a more difficult task, leads to better OSD performance than the model trained directly with a VAD+OSD or OSD objective on the AMI development and evaluation sets and on the CHiME-6 evaluation set. This is especially evident

Table 7: Speaker counting AP (%) achieved by the four considered neural network architectures on the CHiME-6 evaluation set using single-channel features and counting as a training objective.

Counting Model	0-spk	1-spk	2-spk	3-spk	4-spk
LSTM	79.1	69.7	20.5	6.1	**0.002**
CRNN	86.2	73.8	25.4	8.5	**0.003**
TCN	**88.3**	**77.3**	30.0	**12.3**	**0.003**
Transformer	**88.2**	**77.3**	**30.6**	**12.5**	**0.003**

on AMI, where a larger gap between the two models is observed. So, while speaker counting performs poorly on real-world data, it can be convenient to use models trained to perform speaker counting to perform VAD and OSD instead. This may be explained by the fact that speaker count labels provide the model with more information during training than mere OSD labels.

Table 8: VAD and OSD AP (%) achieved by the Transformer-based architecture on the AMI and CHiME-6 development and evaluation sets when using single-channel features and either VAD, OSD, VAD+OSD or counting as the training objective. The values obtained on the development sets are in parentheses.

Method	VAD		OSD	
	AMI	CHiME-6	AMI	CHiME-6
Transformer-VAD	**98.5** (**98.6**)	**94.3** (**93.2**)	n.a.	n.a.
Transformer-OSD	n.a.	n.a.	57.8 (61.0)	50.2 (55.4)
Transformer-VAD+OSD	**98.5** (**98.6**)	**94.3** (93.1)	57.8 (61.0)	49.9 (55.1)
Transformer-Counting	**98.5** (98.5)	**94.3** (**93.2**)	**59.1** (**64.3**)	**50.8** (**55.8**)

6.6. Multichannel Experimental Results

In the following, we select the best model found in Section 6.5, namely the proposed Transformer model trained with a speaker counting objective, and we show how its performance can be improved by employing spatial features along with single-channel features. To do so, we evaluate the IPD, CSIPD and neural network-based spatial features and the early and late fusion schemes described in Section 4 using AMI, CHiME-6 and the proposed synthetic dataset.

In order to allow direct comparison with single-channel results, we adopt the same training strategy as above. Data augmentation is extended to the multichannel scenario by overlapping multichannel audio chunks and being careful, when mixing, in maintaining the array topology (i.e., the first channel is always mixed with the first channel). Training is performed by considering each array separately and using the same FA-based targets as above. Testing is performed, on AMI and CHiME-6, by averaging the predictions made independently for each array across all arrays (i.e., 2 devices for AMI and 6 for CHiME-6).

The IPD and CSIPD features are computed with an STFT window length of 50 ms and the same 10 ms hop-size as single-channel log-Mel features. The corresponding feature vectors, for each microphone pair, are thus of size 801 and 1602, respectively.

Neural network based localization features are extracted using the same Transformer-based architecture as for OSDC, but with $R = 2$ and the modifications outlined in Section 4.2. The network takes CSIPD features relative to most distant microphone pairs with the same STFT window length and hop-size as above, and it outputs $D = 181$ discrete DoAs. It is trained on matched synthetic datasets. More specifically, concerning AMI, we use our synthetic dataset by simulating a circular array instead of the linear one and compute CSIPDs over the 4 pairs obtained by taking opposing microphones in the circular array.

Regarding CHiME-6, we perform training on the Kinect-WSJ2Mix dataset (Sivasankaran et al., 2021) which involves simulated Kinect devices and real CHiME-6 noise and we use CSIPD features between the 3 microphone pairs with largest distance, as explained in Section 4.1. Because Kinect-WSJ2Mix involves at most 2 overlapping speakers while in CHiME-6 up to 4 concurrent speakers can be present, we extend the original data by creating on-the-fly mixtures of up to 4 overlapped speakers and use this newly generated data to train the localization network.

Regarding the experiments performed on the synthetic dataset, we also use CSIPD features between the 3 microphone pairs with largest distance as inputs. Contrary to the AMI and CHiME-6 real-world datasets, in which the localization network is trained on a separate dataset, here we use the same synthetic data for both the OSDC and the localization network.

In addition, to avoid possible domain mismatch between the simulated training dataset for the localization network and the test dataset for the OSDC network, we fine-tune the localization network with the OSDC model by joint optimization with respect to the speaker counting task on the OSDC training dataset. This fine-tuning step is critical to achieve good performance when applying the OSDC network to real-world datasets: for example, on CHiME-6 without fine-tuning the resulting AP is in the order of 50% only. We summarize the datasets used for training the neural localization network, fine-tuning and testing with the back-end OSDC system in Table 9.

Table 9: Datasets used for the neural localization network experiments: training (*train*), fine-tuning (*adapt*) with OSDC back-end and testing (*test*) dataset splits. The total number of hours for each dataset is reported in parenthesis.

Datasets		
Localization Network	OSDC Network	
train	adapt	test
Synthetic (23h)	AMI (81h)	AMI (9h)
Reverberated WSJ-2mix (47h)	CHiME-6 (40.3h)	CHiME-6 (5.2h)
Synthetic (23h)	Synthetic (23h)	Synthetic (4.6h)

In Tables 10 and 11, we report the performance achieved for the VAD and OSD tasks, respectively, with different spatial features, fusion schemes, and numbers of microphone pairs. Microphone pairs are selected as described in Section 4.1, by considering, as the upper bound (*all*), only pairs which add significant spatial diversity, i.e., from 1 to 4 pairs formed by opposing microphones in AMI and from 1 to 3 pairs in CHiME-6 and the synthetic dataset. We also include in the comparison of a single-channel ensemble system with no spatial features, where ensembling is done by averaging the OSDC network outputs over all microphones in the array and a single-channel system trained on beamformed audio using BeamformIt (Anguera et al. (2007)).

Table 10: VAD AP (%) achieved on the AMI, CHiME-6 and synthetic evaluation sets by the Transformer-based architecture trained with a speaker counting objective for different spatial features, fusion schemes, and numbers of microphone pairs (1, 2 or all), as compared to single-channel features only (*None, 1 ch.*), an ensemble of single-channel systems (*None, all ch.*) and a single-channel system + BeamformIt (*None, enh*).

Dataset	Fusion	IPD			CSIPD			Neural	None		
		1	2	all	1	2	all	all	1 ch.	all ch.	enh
AMI	early	98.6	**98.7**	**98.7**	98.6	**98.7**	**98.7**	**98.7**	98.5	98.6	98.5
	late	98.6	**98.7**	**98.7**	98.6	**98.7**	**98.7**	**98.7**			
CHiME-6	early	94.7	94.8	94.8	94.7	94.9	95.1	**95.4**	94.3	94.5	94.3
	late	94.8	95.4	95.4	94.9	95.4	95.4	**95.5**			
Synth	early	96.3	96.8	97.2	96.1	96.4	96.8	**97.5**	96.4	96.6	96.4
	late	96.5	97.2	97.4	96.3	97.1	97.4	**97.5**			

For what concerns VAD performance in Table 10, it can be seen that neural network-based localization features result in on-par or higher performance than the other spatial features, and they outperform single-channel systems by a significant margin on CHiME-6 and the synthetic dataset. Regarding AMI, the AP saturates for most models due to the fact that, as noted previously in Section 6.5, silence is under-represented in the material. An interesting trend which appears on CHiME-6 and synthetic data is that the performance of signal-based spatial features improves when increasing the number of microphone pairs and by using late fusion. Especially on models with late fusion, using more microphones considerably boosts the performance for IPD and CSIPD features. Instead, a smaller improvement is noticeable when early fusion is employed, due to the fact that the size of CSIPD and IPD features grows linearly with the number of pairs but the bottleneck convolutional layer applied in early fusion maps them to a fixed-size representation (384 neurons, as reported in Table 2). Thus some information is inevitably lost in early fusion. On top of that, in late fusion spatial features are available at multiple stages of the architecture.

Table 11: OSD AP (%) achieved on the AMI, CHiME-6 and synthetic evaluation sets by the Transformer-based architecture trained with a speaker counting objective for different spatial features, fusion schemes, and numbers of microphone pairs (1, 2 or all), as compared to single-channel features only (*None, 1 ch.*), ensemble of single-channel systems (*None, all ch.*) and a single-channel system + BeamformIt (*None, enh*).

Dataset	Fusion	IPD			CSIPD			Neural	None		
		1	2	all	1	2	all	all	1 ch.	all ch.	enh
AMI	early	58.1	58.6	**59.4**	57.8	58.4	58.9	**59.3**	57.8	58.6	57.6
	late	58.4	59.5	**60.3**	58.1	59.6	**60.4**	59.7			
CHiME-6	early	51.4	51.5	51.6	51.3	51.4	51.5	**51.8**	50.8	51.2	50.2
	late	51.6	**52.4**	**52.4**	51.7	**52.3**	**52.2**	51.9			
Synth	early	81.8	82.3	82.7	81.6	82.0	82.4	**83.8**	82.4	83.1	82.1
	late	82.8	83.4	84.2	82.9	83.6	**84.4**	**84.3**			

Similar trends can be also observed for OSD performance in Table 11 regarding the number of microphone pairs and early fusion versus late fusion. Notably, neural network-based spatial features are outperformed by signal-based ones on AMI and CHiME-6 when late-fusion is used but reach on-par or top performance when early fusion is employed instead. This suggests that fine-tuning the localization network compensates for the synthetic/real domain mismatch only up to a certain point regarding OSD. It can also be observed that the performance gain achieved by late fusion with respect to early fusion appears modest for neural spatial features, while it is substantial for signal-based ones. This is explained by the fact that neural network-based features are less affected by the aforementioned "bottleneck issue" in early fusion, as they have a more compact size than signal-based ones and, moreover, are jointly fine-tuned with the OSDC system. Again, models with spatial features are able to outperform the single-channel systems and ensembles of single-channel systems. This is notable, as the ensemble is performed using all channels in the array and it comes at the cost of increasing the computational footprint linearly in the number of channels. By contrast, spatial features allow us to boost performance with a smaller increase in computational requirements. The use of beamformed audio degrades OSD performance but not VAD performance with respect to the single-channel only baseline system. This could be explained by the fact that BeamformIt tends to enhance the source with the highest energy and attenuate the rest.

In Tables 12 and 13 we report the counting performance achieved for different spatial features on AMI and CHiME-6, respectively, using two microphone pairs and late fusion. On both datasets, a similar trend can be noticed. On the one hand, neural network based localization features achieve the best figures regarding the 0-spk and 1-spk classes which are the most represented ones. This is in accordance with the VAD results in Table 10 where neural spatial features have in general higher scores. On the other hand, CSIPD and IPD obtain similar or higher AP values for 2 and 3 concurrent speakers. This is in accordance with the OSD results in Table 11. Nonetheless, while systems

based on spatial features are able to substantially increase the speaker counting performance over single-channel systems, the observations made in Section 6.5 are still valid, and reliable speaker counting remains out of reach on real-world data.

Table 12: Speaker counting AP (%) achieved on the AMI evaluation set by the Transformer-based architecture trained with a speaker counting objective for different spatial features, as compared to single-channel features only (*None, 1 ch.*) or an ensemble of single-channel systems (*None, all ch.*).

Spatial Features	0-spk	1-spk	2-spk	3-spk	4-spk
IPD	52.8	88.3	**45.0**	**12.8**	**0.03**
CSIPD	52.9	88.4	**45.1**	**12.7**	**0.03**
Neural	**53.1**	**88.8**	44.9	11.8	**0.03**
None, 1 ch.	50.9	87.2	41.8	11.2	**0.03**
None, all ch.	51.3	87.9	42.4	11.5	**0.03**
None, enh	50.8	87.4	41.6	10.8	**0.03**

Table 13: Speaker counting AP (%) achieved on the CHiME-6 evaluation set by the Transformer-based architecture trained with a speaker counting objective for different spatial features, as compared to single-channel features only (*None, 1 ch.*) or an ensemble of single-channel systems (*None, all ch.*).

Spatial Features	0-spk	1-spk	2-spk	3-spk	4-spk
IPD	89.9	78.8	**32.6**	**12.4**	**0.003**
CSIPD	90.1	78.7	**32.5**	**12.4**	**0.002**
Neural	**90.2**	**79.0**	32.2	11.9	**0.003**
None, 1 ch.	88.2	77.3	30.6	12.5	**0.003**
None, all ch.	90.1	78.4	31.4	11.9	**0.003**
None, enh	88.1	77.4	30.3	11.8	**0.003**

Finally in Figure 5 we use the synthetic dataset to further explain the benefit of spatial features. Using mixtures of two speakers, we report the OSD AP values obtained by the system using single-channel features only versus the ones obtained with late fusion and CSIPD features computed using the 3 microphone pairs with largest distance. The OSD AP performance is plotted against the mean distance of the two speakers from the array and the angle between them as seen from the array. It can be seen that, for the single-channel model, performance degrades to some extent as the speaker distance increases (i.e., colors become darker from bottom to top), but it is largely independent of the angle between the speakers. By contrast, for the model employing spatial features, performance still degrades as the speaker distance increases but at the same time it clearly improves as the angle between the speakers increases (i.e., colors become lighter from left to right). In fact, the AP is significantly boosted for

angles greater than 30 degrees, indicating that spatial features offer complementary information which allows the model to more effectively discriminate frames with overlapped speech.

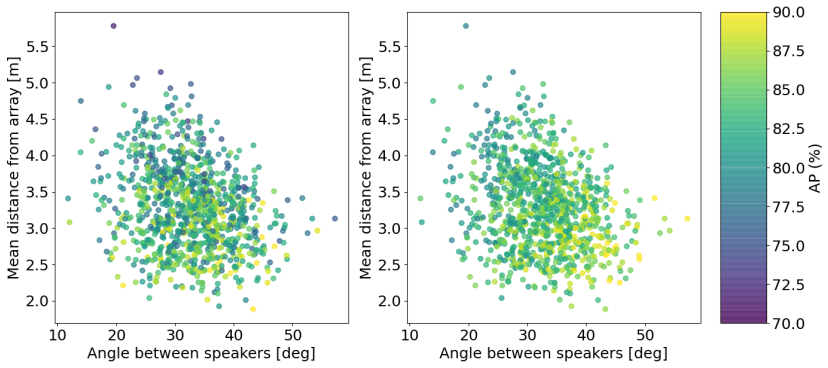

Figure 5: OSD AP (%) achieved on the synthetic evaluation set by the Transformer-based architecture trained with a speaker counting objective as a function of the mean distance of the speakers from the array and the angle between the speakers. Left: single-channel features only. Right: CSIPD spatial features and late fusion.

7. Conclusions

In this paper we studied the problem of performing VAD+OSD and speaker counting on real-world data featuring distant microphone arrays. We focused on neural network based approaches and compared different architectures for the two tasks, on AMI, CHiME-6 and a purposedly developed synthetic dataset. Among the neural networks compared we introduced two novel architectures: one based on TCNs and another based on the Transformer. In parallel we explored the use of spatial features, both signal-based and neural-based, to aid in the VAD+OSD and speaker counting tasks when multiple microphones are available. We conducted an extensive experimental evaluation by comparing the models' computational footprint and VAD, OSD and counting performance on single-channel and multichannel distant speech data. On CHiME-6, our proposed TCN and Transformer-based architectures achieve an absolute improvement in AP of 15% and 16% over previous techniques, respectively. Overall, we found the proposed Transformer-based architecture to be the most promising as it was shown to be able to reach on-par or better results than the other architectures with a significantly lower computational footprint. In general, in comparing VAD+OSD and speaker counting tasks we found that, due to class imbalance, speaker counting performs poorly on real-world data, but, on the other hand, it is desirable to use a speaker counting objective to train a system

to perform VAD+OSD as it is shown to improve OSD. Finally, concerning spatial features, we found that significant further improvements can be obtained by using a late-fusion strategy and by increasing the number of microphone pairs considered. Neural-based spatial features show a clear advantage over signal-based ones for VAD across all datasets, but no spatial feature shows a clear advantage over another for OSD or counting. Future work includes fusing estimates over multiple arrays in a way that favors arrays closer to the speakers and exploits the relative positions and orientations of the arrays whenever they are known, and exploring suitable techniques to counteract the class imbalance problem.

References

Adavanne, S., Politis, A., Virtanen, T., 2018. Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network, in: 26th European Signal Processing Conference (EUSIPCO), pp. 1462–1466.

Andrei, V., Cucu, H., Burileanu, C., 2017. Detecting overlapped speech on short timeframes using deep learning, in: Interspeech, pp. 1198–1202.

Andrei, V., Cucu, H., Burileanu, C., 2019. Overlapped speech detection and competing speaker counting — humans versus deep learning. IEEE Journal of Selected Topics in Signal Processing 13, 850–862.

Andrei, V., Cucu, H., Buzo, A., Burileanu, C., 2015. Counting competing speakers in a timeframe — human versus computer, in: Interspeech, pp. 3399–3403.

Anguera, X., Wooters, C., Hernando, J., 2007. Acoustic beamforming for speaker diarization of meetings. IEEE Transactions on Audio, Speech, and Language Processing 15, 2011–2021.

Arai, T., 2003. Estimating number of speakers by the modulation characteristics of speech, in: ICASSP, pp. II–197.

Ba, J.L., Kiros, J.R., Hinton, G.E., 2016. Layer normalization. Stat 1050, 21.

Bai, S., Kolter, J., Koltun, V., 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint:1803.01271 .

Boakye, K., Vinyals, O., Friedland, G., 2011. Improved overlapped speech handling for speaker diarization, in: Interspeech, pp. 941–944.

Bredin, H., Yin, R., Coria, J.M., Gelly, G., Korshunov, P., Lavechin, M., Fustes, D., Titeux, H., Bouaziz, W., Gill, M.P., 2020. Pyannote.audio: neural building blocks for speaker diarization, in: ICASSP, pp. 7124–7128.

Brutti, A., Omologo, M., Svaizer, P., 2010. Multiple source localization based on acoustic map de-emphasis. EURASIP Journal on Audio, Speech, and Music Processing 2010, 1–17.

Bullock, L., Bredin, H., Garcia-Perera, L.P., 2020. Overlap-aware diarization: Resegmentation using neural end-to-end overlapped speech detection, in: ICASSP, pp. 7114–7118.

Carletta, J., Ashby, S., et al., 2005. The AMI meeting corpus: A pre-announcement, in: International Workshop on Machine Learning for Multimodal Interaction, pp. 28–39.

Chakrabarty, S., Habets, E.A.P., 2017. Broadband DOA estimation using convolutional neural networks trained with noise signals, in: 2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), pp. 136–140.

Charlet, D., Barras, C., Liénard, J.S., 2013. Impact of overlapping speech detection on speaker diarization for broadcast news and debates, in: ICASSP, pp. 7707–7711.

Choi, S., Seo, S., Shin, B., Byun, H., Kersner, M., Kim, B., Kim, D., Ha, S., 2019. Temporal convolution for real-time keyword spotting on mobile devices, in: Interspeech, pp. 3372–3376.

Cornell, S., Omologo, M., Squartini, S., Vincent, E., 2020. Detecting and counting overlapping speakers in distant speech scenarios, in: Interspeech, pp. 3107–3111.

Dehak, N., Kenny, P.J., Dehak, R., Dumouchel, P., Ouellet, P., 2010. Front-end factor analysis for speaker verification. IEEE Transactions on Audio, Speech, and Language Processing 19, 788–798.

Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30.

Diaz-Guerra, D., Miguel, A., Beltran, J.R., 2018. gpuRIR: A Python library for room impulse response simulation with GPU acceleration. arXiv preprint:1810.11359 .

Drude, L., Chinaev, A., Vu, D.H.T., Haeb-Umbach, R., 2014. Source counting in speech mixtures using a variational em approach for complex watson mixture models, in: ICASSP, pp. 6834–6838.

Fujita, Y., Kanda, N., Horiguchi, S., Nagamatsu, K., Watanabe, S., 2019a. End-to-end neural speaker diarization with permutation-free objectives. Interspeech , 4300–4304.

Fujita, Y., Kanda, N., Horiguchi, S., Xue, Y., Nagamatsu, K., Watanabe, S., 2019b. End-to-end neural speaker diarization with self-attention, in: ASRU, pp. 296–303.

Furnon, N., Serizel, R., Illina, I., Essid, S., 2020. Distributed speech separation in spatially unconstrained microphone arrays. arXiv preprint arXiv:2011.00982 .

García-Perera, L.P., Villalba, J., Bredin, H., Du, J., Castán, D., Cristia, A., Bullock, L., Guo, L., Okabe, K., Nidadavolu, P.S., Kataria, S., Chen, S., Galmant, L., Lavechin, M., Sun, L., Gill, M.P., Ben-Yair, B., Abdoli, S., Wang, X., Bouaziz, W., Titeux, H., Dupoux, E., Lee, K.A., Dehak, N., 2020. Speaker detection in the wild: Lessons learned from JSALT 2019, in: Odyssey, pp. 415–422.

Geiger, J., Eyben, F., Schuller, B., Rigoll, G., 2013. Detecting overlapping speech with long short-term memory recurrent neural networks, in: Interspeech, pp. 1668–1672.

Haeb-Umbach, R., Watanabe, S., Nakatani, T., Bacchiani, M., Hoffmeister, B., Seltzer, M.L., Zen, H., Souden, M., 2019. Speech processing for digital home assistants: Combining signal processing with deep-learning techniques. IEEE Signal Processing Magazine 36, 111–124.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification, in: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1026–1034.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 .

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, pp. 448–456.

Kanda, N., Gaur, Y., Wang, X., Meng, Z., Chen, Z., Zhou, T., Yoshioka, T., 2020. Joint speaker counting, speech recognition, and speaker identification for overlapped speech of any number of speakers, in: Interspeech, pp. 36–40.

Kishida, K., 2005. Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments. NII Technical Reports 2005, 1–19.

Knapp, C., Carter, G., 1976. The generalized correlation method for estimation of time delay. IEEE Transactions on Acoustics, Speech, and Signal Processing 24, 320–327.

Kunešová, M., Hrúz, M., Zajíc, Z., Radová, V., 2019. Detection of overlapping speech for the purposes of speaker diarization, in: International Conference on Speech and Computer, pp. 247–257.

Lee, S., Kim, J., Park, J., Hahn, M., 2016. Overlapping speech detection with cluster-based HMM framework, in: 8th International Conference on Signal Processing Systems, pp. 138–141.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft COCO: Common objects in context, in: European Conference on Computer Vision (ECCV), pp. 740–755.

Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., Han, J., 2020. On the variance of the adaptive learning rate and beyond, in: International Conference on Learning Representations.

Luo, Y., Mesgarani, N., 2019. Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation. IEEE/ACM Transactions on Audio, Speech, and Language Processing 27, 1256–1266.

Málek, J., Žďánskỳ, J., 2020. Voice-activity and overlapped speech detection using x-vectors, in: International Conference on Text, Speech, and Dialogue, pp. 366–376.

McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., Sonderegger, M., 2017. Montreal Forced Aligner: Trainable text-speech alignment using Kaldi, in: Interspeech, pp. 498–502.

McCowan, I., Carletta, J., Kraaij, W., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kronenthal, M., Lathoud, G., Lincoln, M., Lisowska Masson, A., Post, W., Reidsma, D., Wellner, P., 2005. The AMI meeting corpus, in: 5th International Conference on Methods and Techniques in Behavioral Research, pp. 137–140.

Medennikov, I., Korenevsky, M., Prisyach, T., Khokhlov, Y., Korenevskaya, M., Sorokin, I., Timofeeva, T., Mitrofanov, A., Andrusenko, A., Podluzhny, I., et al., 2020. Target-speaker voice activity detection: A novel approach for multi-speaker diarization in a dinner party scenario. Interspeech , 274–278.

Nguyen, T.Q., Salazar, J., 2019. Transformers without tears: Improving the normalization of self-attention. arXiv preprint arXiv:1910.05895 .

Ouamour, S., Guerti, M., Sayoud, H., 2008. Pens: a confidence parameter estimating the number of speakers, in: Second ISCA Workshop on Experimental Linguistics, pp. 177–180.

Panayotov, V., Chen, G., Povey, D., Khudanpur, S., 2015. Librispeech: an ASR corpus based on public domain audio books, in: ICASSP, pp. 5206–5210.

Park, D.S., Chan, W., Zhang, Y., Chiu, C.C., Zoph, B., Cubuk, E.D., Le, Q.V., 2019. SpecAugment: A simple data augmentation method for automatic speech recognition, in: Interspeech, pp. 2613–2617.

Pasha, S., Donley, J., Ritz, C., 2017. Blind speaker counting in highly reverberant environments by clustering coherence features, in: 2017 APSIPA Annual Summit and Conference, pp. 1684–1687.

Pavlidi, D., Griffin, A., Puigt, M., Mouchtaris, A., 2012. Source counting in real-time sound source localization using a circular microphone array, in: 2012 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM), pp. 521–524.

Perez, E., Strub, F., De Vries, H., Dumoulin, V., Courville, A., 2018. Film: Visual reasoning with a general conditioning layer, in: 32nd AAAI Conference on Artificial Intelligence, pp. 3942–3951.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlíček, P., Qian, Y., Schwarz, P., Silovský, J., Stemmer, G., Veselý, K., 2011. The Kaldi speech recognition toolkit. Technical Report.

Ryant, N., Bergelson, E., Church, K., Cristia, A., Du, J., Ganapathy, S., Khudanpur, S., Kowalski, D., Krishnamoorthy, M., Kulshreshta, R., Liberman, M., Lu, Y., Maciejewski, M., Metze, F., Profant, J., Sun, L., Tsao, Y., Yu, Z., 2018. Enhancement and analysis of conversational speech: JSALT 2017, in: ICASSP, pp. 5154–5158.

Sajjan, N., Ganesh, S., Sharma, N., Ganapathy, S., Ryant, N., 2018. Leveraging LSTM models for overlap detection in multi-party meetings, in: ICASSP, pp. 5249–5253.

Sivasankaran, S., 2020. Localization guided speech separation. Ph.D. thesis. Université de Lorraine.

Sivasankaran, S., Vincent, E., Fohr, D., 2020. Keyword-based speaker localization: Localizing a target speaker in a multi-speaker environment, in: Interspeech, pp. 2703–2707.

Sivasankaran, S., Vincent, E., Fohr, D., 2021. Analyzing the impact of speaker localization errors on speech separation for automatic speech recognition, in: 28th European Signal Processing Conference (EUSIPCO).

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., Khudanpur, S., 2018. X-vectors: Robust DNN embeddings for speaker recognition, in: ICASSP, pp. 5329–5333.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15, 1929–1958.

Stöter, F.R., 2019. Separation and count estimation for audio sources overlapping in time and frequency separation and estimation of the number of audio signal sources with time and frequency "u overlapping .

Stöter, F.R., Chakrabarty, S., Edler, B., Habets, E.A.P., 2019. CountNet: Estimating the number of concurrent speakers using supervised learning. IEEE/ACM Transactions on Audio, Speech and Language Processing 27, 268–282.

Terpstra, D., Jagode, H., You, H., Dongarra, J., 2009. Collecting performance data with PAPI-C, in: 3rd International Workshop on Parallel Tools for High Performance Computing, pp. 157–173.

Tong, S., Chen, N., Qian, Y., Yu, K., 2014. Evaluating VAD for automatic speech recognition, in: 12th International Conference on Signal Processing (ICSP), pp. 2308–2314.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need, in: 30th International Conference on Neural Information Processing Systems (NIPS), pp. 5998–6008.

Vecchiotti, P., Ma, N., Squartini, S., Brown, G.J., 2019a. End-to-end binaural sound localisation from the raw waveform, in: ICASSP, pp. 451–455.

Vecchiotti, P., Pepe, G., Principi, E., Squartini, S., 2019b. Detection of activity and position of speakers by using deep neural networks and acoustic data augmentation. Expert Systems with Applications 134, 53 – 65.

Vincent, E., Virtanen, T., Gannot, S. (Eds.), 2018. Audio Source Separation and Speech Enhancement. Wiley.

Vipperla, R., Geiger, J.T., Bozonnet, S., Wang, D., Evans, N., Schuller, B., Rigoll, G., 2012. Speech overlap detection and attribution using convolutive non-negative sparse coding, in: ICASSP, pp. 4181–4184.

Walter, O., Drude, L., Haeb-Umbach, R., 2015. Source counting in speech mixtures by nonparametric Bayesian estimation of an infinite Gaussian mixture model, in: ICASSP, pp. 459–463.

Watanabe, S., Delcroix, M., Metze, F., Hershey, J.R. (Eds.), 2017. New Era for Robust Speech Recognition — Exploiting Deep Learning. Springer.

Watanabe, S., Mandel, M., Barker, J., Vincent, E., Arora, A., Chang, X., Khudanpur, S., Manohar, V., Povey, D., Raj, D., Snyder, D., Subramanian, A.S., Trmal, J., Yair, B.B., Boeddeker, C., Ni, Z., Fujita, Y., Horiguchi, S., Kanda, N., Yoshioka, T., Ryant, N., 2020. CHiME-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings, in: 6th International Workshop on Speech Processing in Everyday Environments (CHiME).

Xiao, X., Zhao, S., Zhong, X., Jones, D.L., Chng, E.S., Li, H., 2015. A learning-based approach to direction of arrival estimation in noisy and reverberant environments, in: ICASSP, pp. 2814–2818.

Xu, C., Li, S., et al., 2013. Crowd++: unsupervised speaker count with smartphones, in: 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 43–52.

Yella, S.H., Bourlard, H., 2014. Overlapping speech detection using long-term conversational features for speaker diarization in meeting room conversations. IEEE/ACM Transactions on Audio, Speech, and Language Processing 22, 1688–1700.