Wavesplit: End-to-End Speech Separation by Speaker Clustering

Neil Zeghidour and David Grangier

Abstract—**We introduce Wavesplit, an end-to-end source separation system. From a single mixture, the model infers a representation for each source and then estimates each source signal given the inferred representations. The model is trained to jointly perform both tasks from the raw waveform. Wavesplit infers a set of source representations via clustering, which addresses the fundamental permutation problem of separation. For speech separation, our sequence-wide speaker representations provide a more robust separation of long, challenging recordings compared to prior work. Wavesplit redefines the state-of-the-art on clean mixtures of 2 or 3 speakers (WSJ0-2/3mix), as well as in noisy and reverberated settings (WHAM/WHAMR). We also set a new benchmark on the recent LibriMix dataset. Finally, we show that Wavesplit is also applicable to other domains, by separating fetal and maternal heart rates from a single abdominal electrocardiogram.**

I. Introduction

Source separation is a fundamental problem in machine learning and signal processing, in particular in the ill-posed setting of separating multiple sources from a single mixture. An additional difficulty arises when the sources to separate belong to the *same* class of signals. For example, tasks such as separating overlapped speech, isolating appliance electric consumption from meter reading [1], separating overlapped fingerprints [2], identifying exoplanets in multi-planetary systems from light curves [3] or retrieving individual compounds in chemical mixtures from spectroscopy [4] are particularly difficult as the sources are similar in nature, and as any permutation of them is a correct prediction. This leads to the fundamental *permutation problem* where predicted sources are well separated but inconsistent along time [5]. This situation does not occur when separating different musical instruments from predefined categories [6], [7], or separating speech from non-speech noise [8]. Thus, designing a model that maintains a consistent assignment between the ground-truth sources and the predicted sources is crucial for the tasks with similar sources.

This work precisely aims at separating sources of the same nature from a single mixture. In particular, speech separation aims at isolating individual speaker voices from a recording with overlapping speech [9]. This task is particularly important for public events, conversations and meeting recordings. Research on speech separation spans several decades [9] and it is the most active and competitive field of research in separation [10]–[13]. We therefore introduce our model in the context of this application. Still, to show its generality, we also apply our model to the separation of fetal and maternal heart rate from a single abdominal electrode.

Our approach, Wavesplit, aims at separating novel sources at test time, called open speaker separation in speech, but leverages source identities during training. Specifically, our joint training procedure for speaker identification and speech separation differs from prior research [14]. The training objective encourages identifying instantaneous speaker representations such that (i) these representations can be grouped into individual speaker clusters and (ii) the cluster centroids provide a long-term speaker representation for the reconstruction of individual sources. The extraction of an explicit, long-term representation per source is novel and is beneficial for both speech and non-speech separation. This representation limits inconsistent assignments between ground-truth and predicted sources (source swap), a common type of error for permutation-invariant training (PIT), the dominant approach in speech separation.

Our contributions are six-fold, (i) we leverage training speaker labels but do not need any information about the test speakers beside the mixture recording, (ii) we aggregate information about sources over the whole input mixture which limits channel-swap, (iii) we use clustering to infer source representations which naturally outputs sets, i.e. order-agnostic predictions, (iv) we report state-of-the-art results on the most common speech separation benchmarks, both for clean (WSJ0-2/3mix, Libri2/3mix clean) and noisy settings (WHAM and WHAMR, Libri2/3mix noisy), (v) we analyze the empirical advantages and drawbacks of our method, (vi) we show that our approach is generic and can be applied to non-speech tasks, by separating maternal and fetal heart rate from a single abdominal electrode.

II. Related Work

Single-channel separation takes a single recording with overlapping source signals and predicts the isolated sources. This task is classical in speech processing [9], [15], [16] and has witnessed fast progress recently with supervised neural networks [17]. These models have historically relied on learning time-frequency masks. They divide the input mixture in time-frequency bins (TFB) using a short-term Fourier Transform [18], and identify the source with the maximal energy for each TFB. Source spectrograms can then be produced by masking the TFBs of the other sources, and source signals are later estimated by phase reconstruction [19], [20]. Soft masking variants assign each TFB to multiple speakers with different weights [21].

A. Deep Clustering Approaches

These approaches devise a clustering model for masking: the model learns a latent representation for each TFB such that the distance between TFBs from the same source is lower than the distance between TFBs from different sources. The inference procedure clusters these representations to group TFBs by source [10]. Deep Attractor Networks rely on a similar idea and learn a latent attractor representation for each source such that each TFB is assigned a representation close to their source attractor [22], [23]. [24] also performs clustering at training time by introducing a differentiable unfolded version of k-means. A comparison of methods relying on clustering for frequency masking has been conducted in [25]. Wavesplit also relies on clustering to infer source representations but these representations are not tied to frequency bins and no masking is performed. Instead its representations are learned to (i) predict training speaker identity and (ii) provide conditioning variables to our separation convolutional network.

B. Permutation-Invariant Training

Permutation-Invariant Training (PIT) avoids clustering and predicts multiple masks directly [26]–[28]. The predictions are compared to the ground-truth masks by searching over the permutations of the source orderings. The minimum error over all permutations is used to train the model. PIT acknowledges that the order of predictions and labels in speech separation is irrelevant, i.e. separation is a set prediction problem. Among PIT systems, time domain approaches avoid phase reconstruction and its degradation. They predict audio directly with convolutional [11], [29] or recurrent networks [12]. In that case, PIT compares audio predictions to all permutations of the ground-truth signals. Wavesplit is also a time domain approach but it solves the permutation problem prior to signal estimation: at train time, the latent source representations are ordered to best match the labels prior to conditioning the separation network.

C. Discriminative Speaker Representations

Discriminative speaker representations can be extracted from short speech segments [30], [31] to help separation. Wang et al. [14] extract a representation of the targeted speaker from a clean enrollment sequence and then isolate that speaker in a mixture. Other techniques for separating voices from enrolled speakers have been proposed previously, including speaker beam [32], speaker attention networks [33] or speaker extraction networks [34]. These methods however do not apply to open-set speaker separation, where enrollment data is not available for the test speakers. Nachmani et al. [13] adopt on a similar strategy but can accommodate new sepakers since they rely on a separately trained speaker identification network to represent speakers. [35] proposes to train a network which can both accommodate training data with known and novel speakers, using PIT. Conversely, Wavesplit does not require pre-existing speaker representations but jointly learns to identify and separate sources.

D. Speaker Separation In Constrained Settings

Outside of reconstruction quality, other aspects are important to consider for source separation systems. Separation methods can be required to perform online separation, which is challenging for clustering based methods [36]. Efficiency constraints for inference are also impacting the design of separation systems [37], [38]. Recently, methods which perform separation along other tasks such as diarization and identification have been proposed [39]. Separation in noisy environment is also an important application domain. [40] and [41] explore architectures to address challenging noise settings, including reverberated speech. In this work, we focus on reconstruction quality and leave these efficiency constraints for future work. In noisy settings, we explore additive noise and reverberation and report state-of-the-art results on benchmark datasets.

E. Separation of Fetal and Maternal Heart Rates

Separation of multiple signals from a single sensor is not limited to speech and audio. In this work, we examine separation of fetal and maternal heart rates from abdominal electrocardiograms (ECGs). This technique allows for affordable, non-invasive monitoring of fetal health during pregnancy and labour [42], [43], as an alternative to fetal scalp ECG. Source separation is an intermediate step for detecting fetal heart rate peaks [44], [45]. This work evaluates Wavesplit for separating maternal and fetal heart rate signals.

III. Wavesplit

Wavesplit combines two convolutional subnetworks: the speaker stack and the separation stack (Figure 1). The speaker stack maps the mixture to a set of vectors representing the recorded speakers. The separation stack consumes both the mixture and the set of speaker representations from the speaker stack. It produces a multi-channel audio output with separated speech from each speaker.

The separation stack is classical and resembles previous architectures conditioned on pre-trained speaker vectors [14], or trained with PIT [11]. The speaker stack is novel and constitutes the heart of our contribution. This stack is trained jointly with the separation stack. At training time, speaker labels are used to learn a vector representation per speaker such that the inter-speaker distances are large, while the intra-speaker distances are small. At the same time, this representation is also learned to allow the separation stack to reconstruct the clean signals. At test time, the speaker stack relies on clustering to identify a centroid representation per speaker.

Our strategy contrasts with prior work. Unlike Wang et al. [14], we do not need an enrollment sequence for test speakers since the representation of all speakers is directly inferred from the mixture. With joint training, the speaker representation is not solely optimized for identification but also for the reconstruction of separated speech. In contrast with PIT [26], we condition decoding with a speaker representation valid for the whole sequence. This long-term representation yields excellent performance on long sequences, especially when the relative energy between speakers is varying, see Section IV-B. Our model is less prone to source swap since clustering assigns

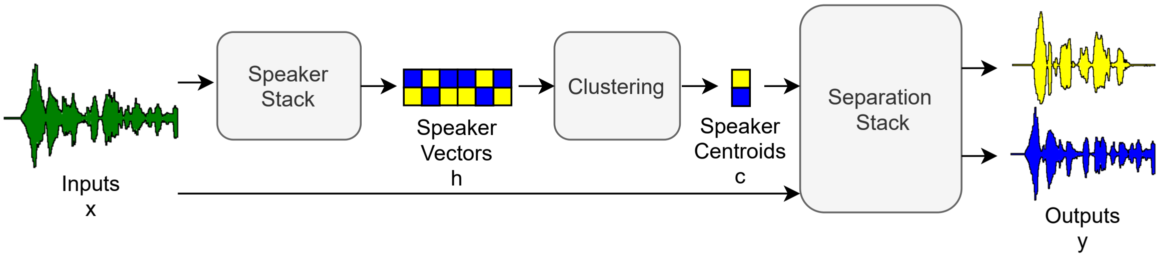

Fig. 1. Wavesplit for 2-speaker separation. The speaker stack extracts speaker vectors at each timestep. The vectors are clustered and aggregated into speaker centroids. The separation stack ingests the centroids and the input mixture to output two clean sources.

a persistent source representation to each predicted source. Still in contrast with PIT, we resolve the permutation ambiguity during training at the level of the speaker representation, i.e. the separation stack is conditioned with speaker vectors ordered consistently with the labels. This does not force the separation stack to choose a latent ordering and allows training this stack with different permutations of the same labels.

A. Problem Setting & Notations

We consider a mixture of N sources. Each single-channel source waveform $i \in [1, N]$ is represented by a continuous vector $y^i \in \mathcal{X}^{1,T}$, with T the length of the sequence. Given a mixture $x = \sum_{i=1}^N y^i$, the source separation task is to reconstruct each y^i.

A separation model f predicts an estimate for each source. Its quality is assessed by comparing its predictions to the reference sources $\{y^i\}_{i=1}^N$ up to a permutation since the source order is arbitrary,

$$Q(\hat{y}, y) = \max_{\sigma \in S_N} \frac{1}{N} \sum_{i=1}^{N} q(\hat{y}^{\sigma(i)}, y^i) \quad \text{where} \quad \forall i, \hat{y}^i = f^i(x). \tag{1}$$

$q(\cdot, \cdot)$ denotes a single-channel reconstruction quality metric and S_N denotes the space of permutations over $[1, N]$. The speaker separation literature typically relies on Signal-to-Distortion Ratio (SDR) to assess reconstruction quality. SDR is the opposite of the log squared error normalized by the energy of the reference signal, $\mathrm{SDR}(\hat{y}, y) = -10 \log_{10} \left(\|y - \hat{y}\|^2 \right) + 10 \log_{10} \left(\|y\|^2 \right)$. Scale-invariant SDR (SI-SDR) considers prediction scale irrelevant and searches over gains [46]. Variants searching over richer signal transforms have also been proposed [47].

B. Model Architecture

Wavesplit is a residual convolutional network with two subnetworks or *stacks*. The first stack transforms the input mixture into a representation of each speaker, while the second stack transforms the input mixture into multiple isolated recordings conditioned on the speaker representation.

The **speaker stack** produces speaker representations at each time step and then performs an aggregation over the whole sequence. Precisely, the speaker stack first maps the input $x = x_{t=1}^T$ into N same-length sequences of latent vectors of dimension d, i.e. $h(x) = \{h^i\}_{i=1}^N$ where $\forall i, h^i \in \mathbb{R}^{T \times d}$. N represents the maximum number of simultaneous speakers targeted by the system, while d is a hyper-parameter selected by cross-validation. Intuitively, h produces a latent representation of each speaker at every time step. It is important to note that h is not required to order speakers consistently across a sequence. E.g. a given speaker Bob could be represented by the first vector h_t^1 at time t and by the second vector $h_{t'}^2$ at a different time t'. At the end of the sequence, the aggregation step groups all vectors by speaker and outputs N summary vectors for the whole sequence. K-means clustering performs this aggregation at inference [48] and returns the centroids of the N identified clusters,

$$c = \{c_i\}_{i=1}^N = \mathrm{kmeans}(\{h_t^i\}_{i,t}; N).$$

In the following, we refer to the local vectors h_t^i as the *speaker vectors*, and to the vectors c_i as the *speaker centroids*. During training, clustering is not used. Speaker centroids are derived by grouping speaker vectors by speaker identity, relying on the speaker training objective described in Section III-C.

The **separation stack** maps the mixture x and the speaker centroids c into an N-channel signal $\hat{y}$,

$$\hat{y} = f(x, c) = f(x, \mathrm{kmeans}(h(x); N)).$$

Inspired by Luo and Mesgarani [11], we rely on a **residual convolutional architecture** for both stacks. Each residual block in the speaker stack composes a dilated convolution dconv [49], a non-linearity nl and layer normalization lnorm [50],

$$x_{l+1} = x_l + \mathrm{lnorm}(\mathrm{nl}(\mathrm{dconv}(x_l))).$$

We use parametric rectified linear units [51] for nl after experimenting with multiple alternatives. The last layer of the speaker stack applies Euclidean normalization to the speaker vectors.

The residual blocks of the **separation stack** are conditioned by the speaker centroids relying on FiLM, Feature-wise Linear Modulation [52],

$$x_{l+1} = x_l + \mathrm{lnorm}(\mathrm{nl}(a * \mathrm{dconv}(x_l) + b))$$

where $a = \mathrm{lin}(c)$ and $b = \mathrm{lin}'(c)$ are different linear projections of c, the concatenation of the speaker centroids. Section IV-F shows the advantage of FiLM conditioning over classical bias only conditioning, $a = 1$ [53]. We learn distinct parameters for each layer for all parametric functions, i.e. dconv, nl, lnorm, lin and lin'.

C. Model Training Objective

Model training addresses two objectives: (i) it learns speaker vectors which can be clustered by speaker identity into well separated clusters; (ii) it optimizes the reconstruction of the separated signals from aggregated speaker vectors.

Wavesplit assumes the training data is annotated with speaker identities from a finite set of M training speakers but does not require any speaker annotation at test time. Speaker identities are labeled in most separation datasets, including Wall Street Journal variants [10], [54], [55], meeting recordings [56] or cocktail party recordings [57], but mostly unused in the source separation literature. Wavesplit exploits this information to build an internal model of each source and improve long-term separation.

The **speaker vector objective** encourages the speaker stack outputs to have small intra-speaker and large inter-speaker distances. From an input x with target signals $\{y^i\}_{i=1}^N$ and corresponding speakers $\{s_i\}_{i=1}^N \in [1, M]^N$, the speaker loss favors correct speaker identification at each time step t, i.e.

$$\mathcal{L}_{\text{speaker}}(x, \{s_i\}_{i=1}^N) = \sum_{t=1}^{T} \min_{\sigma \in S_N} \sum_{i=1}^{N} \ell_{\text{speaker}}(h_t^{\sigma(i)}, s_i)$$

where ℓ_{speaker} defines a loss function between a vector of $\mathbb{R}^d$ and a speaker identity of $[1, M]$. The minimum over permutations expresses that each identity should be identified at each time step, in any arbitrary order. The best permutation (argmin) at each time-step is used to re-order the speaker vectors in an order consistent with the training labels. This allows averaging the speaker vectors originating from the same speaker at training time. This makes optimization simpler compared to work relying on k-means during training [24]. This permutation per time-step differs from PIT [28]: we do not require the model to pick a single ordering over the output sources, as eventual source swaps at the level of speaker vectors will be corrected by k-means. Moreover, the separation stack is trained from different permutations of the same labels.

Three alternative are explored for ℓ_{speaker}. All three maintain an embedding table over training speakers $E \in \mathbb{R}^{M \times d}$. First, $\ell_{\text{speaker}}^{\text{dist}}$ is a distance objective. This loss favors small distances between a speaker vector and the corresponding embedding while enforcing the distance between different speaker vectors at the same time-step to be larger than a margin of 1,

$$\ell_{\text{speaker}}^{\text{dist}}(h_t^j, s_i) = \|h_t^j - E_{s_i}\|^2 + \sum_{k \neq j} \max(0, 1 - \|h_t^j - h_t^k\|^2) \tag{2}$$

Second, $\ell_{\text{speaker}}^{\text{local}}$ is a local classifier objective which discriminates among the speakers present in the sequence. It relies on the log softmax over the distances between speaker vectors and embeddings,

$$\ell_{\text{speaker}}^{\text{local}}(h_t^j, s_i) = d(h_t^j, E_{s_i}) + \log \sum_{k=1}^{N} \exp\left(-d(h_t^j, E_{s_k})\right) \tag{3}$$

where $d(h_t^j, E_{s_i}) = \alpha\|h_t^j - E_{s_i}\|^2 + \beta$ is the squared Euclidean distance rescaled with learned scalar parameters $\alpha > 0, \beta$. Finally, the global classifier objective $\ell_{\text{speaker}}^{\text{global}}$ is similar, except that the partition function is computed over all speakers in the training set, i.e.

$$\ell_{\text{speaker}}^{\text{global}}(h_t^j, s_i) = d(h_t^j, E_{s_i}) + \log \sum_{k=1}^{M} \exp\left(-d(h_t^j, E_k)\right). \tag{4}$$

This loss corresponds to the cross-entropy loss of a classifier using $d(h, E_i)$ as local potentials (logits). We use $\mathcal{L}_{\text{speaker}}$ to update the speaker stack, as well as the speaker embedding table. The **reconstruction objective** aims at optimizing the separation quality, as defined in Eq. (1).

$$\mathcal{L}_{\text{reconstr}}(f(x, c), y) = \frac{1}{N} \sum_{i=1}^{N} \ell_{\text{reconstr}}(f(x, c)^i, y^i). \tag{5}$$

Contrasting with PIT [26], this expression does not require searching over the space of permutations since the centroids $c = \{c_i\}_{i=1}^N$ are consistent with the order of the labels $\{y^i\}_{i=1}^N$ as explained above. For ℓ_{reconstr}, we use negative SDR with a clipping τ to limit the influence of the best training predictions [58], $\ell_{\text{reconstr}}(\hat{y}, y) = -\min(\tau, \text{SDR}(\hat{y}, y))$. Inspired by [13], we compute $\mathcal{L}_{\text{reconstr}}$ at each layer of the separation stack, and use the average over all layers as our reconstruction loss.

We consider different forms of **regularization** to improve generalization to new speakers. At training time, we add Gaussian noise to speaker centroids, we replace full centroids with zeros (speaker dropout) and we replace some centroids with a linear combination with other centroids from the same training batch (speaker mixup). Speaker dropout and mixup are inspired by dropout [59] and mixup [60]. Note that regardless of the number of speakers in a sequence, speaker dropout removes at most one centroid, such that the separation task is not ambiguous ($N-1$ speaker-source assignments are sufficient to reconstruct the Nth). Finally, we favor well separated embeddings for the training speakers with entropy regularization [61], $\ell_{\text{reg}} = -\sum_i \min_{j \neq i} \log \|E_i - E_j\|$. Section IV-F analyses the benefits of regularization.

D. Training Algorithm

Model training optimizes the weighted sum of $\mathcal{L}_{\text{speaker}}$ and $\mathcal{L}_{\text{reconstr}}$ with Adam [62]. We train on mini batches of fixed-size windows. Wavesplit performs well for a wide-range of window sizes starting at 750ms, unlike most PIT approaches [11]–[13] that require longer segments ($\sim$ 4s). The training set is shuffled at each epoch and a window starting point is uniformly sampled each time a sequence is visited. This sampling gives the same importance to each sequence regardless of its length. This strategy is consistent with the averaging of per-sequence SDR used for evaluation. We replicate each training sequence for all permutations of the target signals to avoid over-fitting a specific ordering. Replication, windowing and shuffling are not applied to validation or test data.

E. Data Augmentation with Dynamic Mixing

Separation benchmarks like WSJ0-2mix [10] create a standard split between train, valid and test sequences and then generate a finite set of input mixtures by summing specific clean signals with specific weights (gains). As an orthogonal contribution to our model, we consider creating training mixtures dynamically. Our training augmentation creates new examples indefinitely by sampling random windows of training recordings to be summed after applying random gains. A similar method has been used in music source separation [63], [64]. This simple method brings systematic improvements, which advocate for the use of this training scheme when generating mixtures on the fly is not costly. We also experiment without augmentation to isolate the impact of Wavesplit alone.

IV. Experiments & Results

Most experiments are performed on the speaker separation dataset [65] built from the LDC WSJ-0 dataset [66] as introduced in [10]. We rely on the 8kHz version of the data, with 2 or 3 concurrent speakers. This setting is a de-facto benchmark for open-speaker source separation and we compare our results to alternative methods. Table I reports the dataset statistics. Additionally, we perform experiments in noisy settings. We rely on WHAM! with urban noise [54] and WHAMR! with noise and reverberation [55]. These datasets are derived from WSJ0-2mix and have identical statistics. We further evaluate variants of Wavesplit with different loss functions and architectural alternatives. We conduct an error analysis examining a small fraction of sequences with a strong negative impact on overall performance. We also perform experiments on the recently released LibriMix dataset [67]. We also rely on the 8 kHz version of the data. Like for WSJ-0, we evaluate our model in clean and noisy settings with 2 or 3 concurrent speakers. The statistics of this larger dataset are also in Table I. Finally, we show results on a fetus/mother heart rate separation task.

TABLE I
WSJ0-mix and LibriMix statistics.

Dataset		train	valid	test
WSJ0-2mix	# examples	20k	5k	3k
	# speakers	101		18
	mean length	5.4 sec	5.5 sec	5.7 sec
WSJ0-3mix	# sequences	20k	5k	3k
	# speakers	101		18
	mean length	4.9 sec	4.9 sec	5.2 sec
Libri2mix	# examples	51k	3k	3k
	# speakers	921	40	40
	mean length	15.0 sec	13.2 sec	13.2 sec
Libri3mix	# sequences	34k	3k	3k
	# speakers	921	40	40
	mean length	15.5 sec	13.2 sec	13.2 sec

Our evaluation uses signal-to-distortion ratio (SDR) and scale-invariant SDR (SI-SDR) [46], [47], see Section III-A. SDR is measured using the standard MIR-eval library [68]. Like prior work, we report results as *improvements*, i.e. the metric obtained using the system output minus the metric obtained by using the input mixture as the prediction. We provide recordings processed by our system on a public webpage [1] as well as in supplementary material.

A. Hyperparameter Selection

Preliminary experiments on WSJ0-2mix [10] drove our architecture choices for subsequent ones. Both stacks have a latent dimension of 512 and the dilated convolutions have a kernel size of 3 without striding, therefore all layers preserve the temporal resolution of the input signal and no upsampling is necessary at the output. The dilation factor varies with depth. The speaker stack is 14-layer deep and dilation grows exponentially from 2^0 to 2^{13}. The separation stack has 40 layers with the dilation pattern from [53], i.e. $\delta_l = 2^{l \bmod 10}$. Every 10 layers, the dilation is reset to 1 allowing multiple fine-to-coarse-to-fine interactions across the time axis.

For training, we validated a learning rate of $1\mathrm{e}{-3}$ in $[1\mathrm{e}{-3}, 2\mathrm{e}{-3}, 3\mathrm{e}{-3}]$ and a speaker loss weight of 2 in $[1, 2, 5]$. For regularization, we validated a distance regularization weight at 0.3 in $[0, 0.2, 0.3, 0.5]$ and a Gaussian noise with standard deviation at 0.2 in $[0, 0.1, 0.2, 0.3]$. We use a speaker dropout rate of 0.4 (picked in $[0, 0.2, 0.4, 0.6]$) and a speaker mixup rate of 0.5 (picked in $[0, 0.5, 1]$). The clipping threshold on the negative SDR loss was validated at 30 for clean data and 27 for noisy data within $[22, 27, 30]$. For $\mathcal{L}_{\mathrm{speaker}}$, we found the global classifier to be the most effective, see Section IV-F.

B. Clean Settings

WSJ0-2mix/3-mix is the de facto benchmark for separation. Table II reports the results for 2 and 3 simultaneous speakers. In both cases, Wavesplit outperforms alternatives and dynamic mixing further increases this advantage. For instance, on WSJ0-2mix we report 21.0 ΔSI-SDR compared to 20.1 for the recent gated dual path RNN [13]. This number improves to 22.2 with dynamic augmentation.

On WSJ0-2mix, Table III analyses the error distribution. The test set has more sentences with poor ΔSDR (< 10) compared to validation, 5.6% versus 0.9%. Unlike test data, validation data contains the same speakers as the training set and we observe that test examples with low ΔSDR are sequences where both speakers are close to the same training speaker identity. We report that 77% of the poor performing test sequences have confusing speakers, which refers to cases where the same training speaker embedding is the nearest neighbor of more than 10% of the frames, for both speaker vector channels. Our oracle takes the predicted signal and, for each frame, it permutes the predicted samples across sources to best match the ground-truth signal. The accuracy of this oracle frame-wise permutation shows that most of the errors are source assignment errors.

WSJ0-2mix recordings have a single dominant speaker, i.e. the same speaker stays the loudest throughout the whole recording. PIT might implicitly rely on this bias to address the source assignment ambiguity and we evaluate robustness to change in dominant speaker on long sequences. We concatenate test sequences with the same pair of speakers, for length up to 10 times the original length. Unlike the training data, the loudest speaker changes between each concatenated sequence. For PIT models, we retrain Conv-TasNet [74] and take a pre-trained Dual-Path RNN model [75]. Table IV shows the advantage of Wavesplit, which explicitly models sources in a

[1] https://soundcloud.com/wavesplitdemo/sets

TABLE II
SI-SDR AND SDR IMPROVEMENTS (DB) ON WSJ0-2MIX AND WSJ0-3MIX.

Model	2 speakers		3 speakers	
	ΔSI-SDR	ΔSDR	ΔSI-SDR	ΔSDR
Deep Clustering [69]	10.8	–	7.1	–
uPIT-blstm-st [27]	–	10.0	–	7.7
Deep Attractor Net. [22]	10.5	–	8.6	8.9
Anchored Deep Attr. [23]	10.4	10.8	9.1	9.4
Grid LSTM PIT [28]	–	10.2	–	–
ConvLSTM-GAT [70]	–	11.0	–	–
Chimera++ [25]	11.5	12.0	–	–
WA-MISI-5 [20]	12.6	13.1	–	–
blstm-TasNet [71]	13.2	13.6	–	–
Conv-TasNet [11]	15.3	15.6	12.7	13.1
Conv-TasNet+MBT [72]	15.5	15.9	–	–
DeepCASA [73]	17.7	18.0	–	–
FurcaNeXt [29]	–	18.4	–	–
DualPathRNN [12]	18.8	19.0	–	–
Gated DualPathRNN [13]	20.1	–	16.9	–
Wavesplit	**21.0**	**21.2**	**17.3**	**17.6**
Wavesplit + Dynamic mixing	**22.2**	**22.3**	**17.8**	**18.1**

TABLE III
ERROR ANALYSIS ON WSJ0-2MIX.

	valid		test	
ΔSDR Split	< 10	≥ 10	< 10	≥ 10
Examples %	0.9	99.1	5.6	94.4
Confusing spkr %	60.7	5.5	77.2	12.4
Mean ΔSDR	4.3	22.4	3.9	22.2
Oracle ΔSDR	17.4	22.9	18.8	22.9

TABLE IV
ΔSDR (DB) ON LONG SEQUENCES.

Model	Sequence Length		
	$\times$1	$\times$4	$\times$10
Conv-TasNet	15.6	13.6	14.0
DualPathRNN	19.1	17.3	16.9
Wavesplit	**21.2**	**20.2**	**20.0**

time-independent fashion rather than relying on implicit rules that exploit biases in the training data. This is remarkable since Wavesplit is trained on 1s long windows compared to longer 4s windows for both PIT models.

C. Noisy and Reverberated Settings

WSJ0-2mix was recorded in clean conditions and noisy variants have been introduced to devise more challenging use cases. WHAM! [54] adds noise recorded in public areas to the mixtures. As the model should only predict clean signals, it cannot exploit the fact that predicted sources should sum to the input signal. WHAMR! [55] adds the same noise, but also reverberates the clean signals. This task is even harder as the model should predict clean signals without reverberation, i.e. jointly addressing denoising, dereverberation and source separation. We adapted dynamic mixing for these datasets. For WHAM!, we also sampled a gain for the noise, and combined it to reweighted clean signals to generate noisy mixtures on the fly. We similarly remixed WHAMR!, except that we reweighted reverberated signals with noise. Table V shows that our model outperforms previous work by a substantial margin on both datasets: e.g. our accuracy on WHAMR! is comparable to results on clean inputs (WSJ0-2mix) prior to [25].

D. Large scale experiments on LibriMix

We train Wavesplit on the newly released dataset LibriMix [67], which contains artificial mixtures of utterances from Librispeech [77], in four conditions (2/3 speakers, clean or noisy). We train on the *train-360* subset of LibriMix to allow a fair comparison with the baselines from [67], which are trained in the same conditions. Like for WSJ-0, we rely on the 8 kHz version of the data. LibriMix is a good testbed for Wavesplit, as its training set contains significantly more speakers than WSJ0-2/3mix (921 in *train-360* against 101 in the training set of WSJ0-2/3mix) which improves the robustness of the speaker stack. Table VI reports the results on all four conditions. The baselines are a Conv-TasNet model, as well as frequency masking oracles, Ideal Ratio Mask (IRM) and Ideal Binary Mask (IBM). Wavesplit significantly outperforms Conv-TasNet in all conditions, and even consistently outperforms oracle ideal masks. Moreover, the results are in the same range as the equivalent condition in WSJ0-2/3mix and WHAM!, which again confirms the robustness of the method across datasets.

We also report an error analysis on Librimix in Table VIII. Test failure cases with ΔSDR< 10 are rare, with 1.3% of the sequences. Like for WSJ0, source permutations are the most common source of failure as shown by the good oracle result. For most of these failed sequences, the present speakers are close with 60% confusing speaker cases.

Librimix also allows us to measure the stability of the speaker representation for novel speakers since the 40 test speakers are not seen during training but appear in multiple test sequences. On the 3k sentence test set, our model outputs 6k speaker centroids. For each centroid c, 94.3% of the 40 nearest centroids from other sequences correspond to the same identity (the least frequent speaker appears 45 times, so this metric optimum is 100%), see Table IX (a). We also measure if we identify the correct pair of speakers at each test time step by classifying the frame-wise speaker vectors h using a nearest neighbor

TABLE V
SI-SDR AND SDR IMPROVEMENTS (dB) ON WHAM! AND WHAMR!.

Model	WHAM!		WHAMR!	
	ΔSI-SDR	ΔSDR	ΔSI-SDR	ΔSDR
Conv-TasNet [55], [76]	12.7	–	8.3	–
Learnable fbank [76]	12.9	–	–	–
BLSTM-TasNet [55]	12.0	–	9.2	–
Gated DualPathRNN [13]	15.2	–	12.2	–
Wavesplit	15.4	15.8	12.0	11.1
Wavesplit + Dynamic mixing	**16.0**	16.5	**13.2**	12.2

TABLE VI
SI-SDR AND SDR IMPROVEMENTS (dB) ON LIBRIMIX.

Model	Libri2mix				Libri3mix			
Condition	clean		noisy		clean		noisy	
	ΔSI-SDR	ΔSDR	ΔSI-SDR	ΔSDR	ΔSI-SDR	ΔSDR	ΔSI-SDR	ΔSDR
Conv-TasNet [67]	14.7	–	12.0	–	12.1	–	10.4	–
IRM (oracle) [67]	12.9	–	12.0	–	13.1	–	12.6	–
IBM (oracle) [67]	13.7	–	12.6	–	13.9	–	13.3	–
Wavesplit	19.5	20.0	15.1	15.8	15.8	16.3	13.1	13.8
Wavesplit + Dynamic mixing	**20.5**	20.9	**15.2**	15.9	**17.5**	18.0	**13.4**	14.1

TABLE VII
SI-SDR IMPROVEMENTS (dB) ON SPARSE-LIBRIMIX FOR THE IDEAL RATIO MASK (IRM), CONV-TASNET (CTN) AND WAVESPLIT (WS). IN **BOLD** ARE THE BEST RESULTS PER-DATASET AND PER-OVERLAP, EXCLUDING THE IRM ORACLE WHICH ACCESSES THE LABELS.

Overlap	Libri2mix (clean)			Libri2mix (noisy)			Libri3mix (clean)			Libri3mix (noisy)		
	IRM [67] (*oracle*)	CTN [67]	WS	IRM [67] (*oracle*)	CTN [67]	WS	IRM [67] (*oracle*)	CTN [67]	WS	IRM [67] (*oracle*)	CTN [67]	WS
0%	*43.7*	**31.9**	40.8	*16.1*	**14.5**	12.9	*44.2*	**24.8**	16.9	*18.7*	**13.0**	10.6
20%	*19.6*	20.0	**23.6**	*14.7*	13.9	**14.7**	*18.1*	15.8	**18.1**	*15.6*	12.1	**13.8**
40%	*16.2*	17.6	**21.8**	*13.8*	13.2	**15.1**	*16.4*	14.4	**17.9**	*14.9*	11.7	**13.9**
60%	*14.9*	16.3	**21.1**	*13.3*	12.7	**15.5**	*15.5*	13.8	**17.6**	*14.4*	11.5	**14.2**
80%	*14.2*	15.7	**20.3**	*13.0*	12.5	**15.3**	*14.6*	13.1	**17.4**	*13.9*	11.0	**14.2**
100%	*13.8*	15.3	**19.9**	*12.7*	12.2	**15.0**	*14.3*	12.5	**16.8**	*13.6*	10.7	**13.7**

TABLE VIII
ERROR ANALYSIS ON LIBRIMIX (2 SPEAKERS, CLEAN).

	valid		test	
ΔSDR Split	< 10	≥ 10	< 10	≥ 10
Examples %	2.5	97.5	1.3	98.7
Confusing spkr %	25.3	5.7	60.0	5.6
Mean ΔSDR	6.8	21.2	3.3	21.0
Oracle ΔSDR	12.9	21.7	15.7	21.4

classifier relying on the $6,000-2$ centroids c from the other sequences, see Table IX (b).

Like WSJ0-2/3mix and WHAM(R)!, LibriMix composes artificial mixtures from single speaker segments, with a constant overlap between speakers. A more realistic setting is proposed by the SparseLibriMix dataset [67], which simulates sparsely overlapping mixtures, in which speakers speak at the same time for only a proportion of the total utterance length. To simulate variable overlap we use an additional data augmentation technique for dynamic mixing: each source segment has a subsegment randomly masked with zeroes to simulate silence. Table VII reports results for the Ideal Ratio Mask (IRM), Conv-TasNet and Wavesplit on the sparse versions of LibriMix datasets. Even though without overlap Wavesplit underperforms Conv-TasNet and the IRM in the three-speaker setting, as overlap increases Wavesplit outperforms Conv-TasNet consistently and matches or outperforms the IRM when the overlap proportion is above 60-80%.

E. Handling a variable number of speakers

Wavesplit assumes that we know the number of speakers beforehand, as it defines the number of sources predicted by the system. A model that can operate on a varying number of speakers [13] is more flexible. Rather than separating the tasks of counting sources and separating them, we propose a simple strategy: to separate up to N speakers, we train a model that produces N sources, randomly masking some sources during training (as done for SparseLibriMix, see Section IV-D), to simulate a variable number of sources. We expect that when the model is given $N' < N$ sources, it will output N' active sources and $N-N'$ silent sources. To verify this hypothesis, we use a model trained on SparseLibri3mix (clean) and evaluate it on Libri2mix. As the model produces three sources, we automatically select the ones with high energy as active sources, while removing the silent one. By doing so, the model gets a 18.4 ΔSDR, which shows the robustness of a model trained on 3 speakers when evaluated on 2. Moreover, the relative gain of the predicted source detected as silent compared to the ones

TABLE IX
STABILITY OF REPRESENTATIONS OF THE TEST SPEAKERS ON LIBRIMIX (2 SPEAKERS, CLEAN).

(a) Nearest neighbor centroid statistics	
Num. neighbors	Same speaker
1	98.4%
10	96.9%
20	95.8%
40	94.3%
(b) Nearest neighbor frame-wise identification	
Num. neighbors	Accuracy
1	96.6%

TABLE X
ABLATION ON WSJ0-2MIX.

Model	ΔSDR (dB)
Base model	21.2
w/ distance loss	19.3
w/o FiLM	20.6
w/o distance reg.	20.6
w/o speaker dropout	20.6
w/o speaker mixup	19.6

detected as active is on average of -16.2dB, which shows that the model actually generates silence for missing sources.

F. Ablation Study

Table X compares the base result (no dynamic mixing) obtained with the global classifier loss, Eq. (4), with the distance loss, Eq. (2). Although this type of loss is common in distance learning for clustering [25], the global classifier reports better results. We also experimented with the local classifier loss, Eq. (3), which yielded slower training and worse genereralization. Table X also reports the advantage of multiplicative FiLM conditioning compared to additive conditioning [53]. Not only SDRs are better but FiLM allows using a higher learning rate and yields faster training. Table X also shows the benefit of regularizing the speaker representation.

G. Separation of Maternal and Fetal Electrocardiograms

Wavesplit separation method can be applied beyond speech. Electrocardiogram (ECG) reports voltage time series of the electrical activity of the heart from electrodes placed on the skin. During pregnancy, ECG informs about the function of the fetal heart but maternal and fetal ECG are mixed. We aim at separating these signals from a single noisy electrode recording. We use the FECGSYNDB data [78] which simulates noisy abdominal ECG measurements of pregnancies in varying conditions (e.g. fetal movement, uterine contraction) over 34 electrode locations, each recording being 5 minutes at 250Hz. The database contains 10 pregnancies, that differ by the intrinsic characteristics of the mother and the fetus. We use 6 pregnancies for training, 2 for validation, and 2 for testing. Each sample of an electrode provides the noisy ECG mixture and ground-truth for maternal and fetal ECG. All 34 electrodes are given independently to the model, without location information. We train a single model independently of the electrode's location and rely on the same architecture as in our speech experiments, only validating regularization parameters. In this context, the source/speaker stack learns a representation of a mother and its fetus to condition the separation stack. Wavesplit, Dual-Path RNN, and Conv-TasNet are trained on the same data. Table XI illustrates the advantage of Wavesplit on this task.

TABLE XI
SI-SDR AND SDR IMPROVEMENTS (DB) ON FECGSYNDB (FETAL/MATERNAL ECG).

Model	ΔSI-SDR	ΔSDR
Conv-TasNet	11.4	11.9
DualPathRNN	11.4	11.4
Wavesplit	**12.3**	**14.4**

Figure 2 provides an example signal of Wavesplit applied to maternal/fetal heart rate separation. The input mixture (left) shows that the signal is almost indistinguishable from noise outside of peaks. However, our model extracts both the mother (center) and fetal (right) signal with high accuracy.

V. CONCLUSIONS

We introduce Wavesplit, a neural network for source separation. From the input mixed signal, our model extracts a representation for each source and estimates the separated signals conditioned on the inferred representations. Contrary to prior work, we learn both tasks jointly in an end-to-end manner, optimizing the reconstruction of the separated signals. For each mixed signal, the model learns to predict local representations which can be aggregated into a consistent representation for each source via clustering. Clustering is well suited for separation as it naturally represents a set of sources without arbitrarily ordering them. Separation with Waveplit relies on a single consistent representation of each source regardless of the input signal length. This is advantageous on long recordings, as shown by our experiments on speech separation. For this competitive application, our model redefines the state-of-the-art on standard benchmarks (WSJ-0-mix and LibriMix), both in clean and noisy conditions. We also report the benefits of Wavesplit on fetal/maternal heart rate separation from electrocardiograms. These results open perspectives in other separation domains, e.g. light curves in astronomy, electrical energy consumption, or spectroscopy.

ACKNOWLEDGMENTS

The authors are grateful to Adam Roberts, Chenjie Gu, Raphael Marinier and Olivier Teboul for their advice on model implementation. They are also grateful to Jonathan Le Roux, John Hershey, Richard F. Lyon, Norman Casagrande and Olivier Pietquin for their help navigating speech separation prior work. The authors also thank Fernando Andreotti and Joachim Behar for the FECGSYNDB dataset.

REFERENCES

[1] D. Murray, L. Stankovic, and V. Stankovic, "An electrical load measurements dataset of united kingdom households from a two-year longitudinal study," *Scientific data*, vol. 4, no. 1, pp. 1–12, 2017.

[2] Y. I. Shehu, A. Ruiz-Garcia, V. Palade, and A. James, "Sokoto coventry fingerprint dataset," *arXiv preprint arXiv:1807.10609*, 2018.

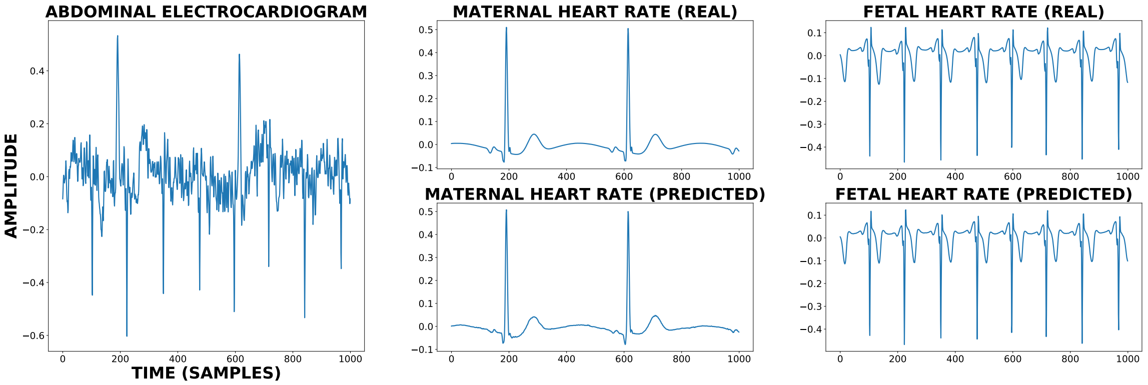

Fig. 2. Example of separation of maternal and foetal heart rate from a simulated abdominal electrode on the FECGSYNDB test set.

[3] C. J. Shallue and A. Vanderburg, "Identifying exoplanets with deep learning: A five-planet resonant chain around kepler-80 and an eighth planet around kepler-90," *The Astronomical Journal*, vol. 155, no. 2, p. 94, 2018.

[4] I. Toumi, S. Caldarelli, and B. Torrésani, "A review of blind source separation in nmr spectroscopy," *Progress in nuclear magnetic resonance spectroscopy*, vol. 81, pp. 37–64, 2014.

[5] C. Weng, D. Yu, M. L. Seltzer, and J. Droppo, "Deep neural networks for single-channel multi-talker speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, 2015.

[6] P. Seetharaman, G. Wichern, S. Venkataramani, and J. L. Roux, "Class-conditional embeddings for music source separation," 2019, pp. 301–305.

[7] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation," Dec. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1117372

[8] P. C. Loizou, *Speech enhancement: theory and practice*. CRC press, 2013.

[9] E. Vincent, T. Virtanen, and S. Gannot, *Audio Source Separation and Speech Enhancement*. John Wiley Sons, Ltd, 2018.

[10] J. R. Hershey, Z. Chen, J. L. Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *ICASSP*. IEEE, 2016, pp. 31–35.

[11] Y. Luo and N. Mesgarani, "Conv-tasnet: Surpassing ideal time-frequency magnitude masking for speech separation," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[12] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation," *arXiv preprint arXiv:1910.06379*, 2019.

[13] E. Nachmani, Y. Adi, and L. Wolf, "Voice separation with an unknown number of multiple speakers," *arXiv preprint arXiv:2003.01531*, 2020.

[14] Q. Wang, H. Muckenhirn, K. W. Wilson, P. Sridhar, Z. Wu, J. R. Hershey, R. A. Saurous, R. J. Weiss, Y. Jia, and I. Lopez-Moreno, "Voicefilter: Targeted voice separation by speaker-conditioned spectrogram masking," *CoRR*, vol. abs/1810.04826, 2018.

[15] S. T. Roweis, "One microphone source separation," in *Advances in neural information processing systems*, 2001, pp. 793–799.

[16] O. Yilmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Transactions on signal processing*, vol. 52, no. 7, pp. 1830–1847, 2004.

[17] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.

[18] D. Williamson, *Discrete-time Signal Processing: An Algebraic Approach*, ser. Advanced Textbooks in Control and Signal Processing. Springer, 2012.

[19] D. Griffin and Jae Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1984.

[20] Z. Wang, J. L. Roux, D. Wang, and J. R. Hershey, "End-to-end speech separation with unfolded iterative phase reconstruction," in *INTERSPEECH*. ISCA, 2018, pp. 2708–2712.

[21] S. Araki, S. Makino, H. Sawada, and R. Mukai, "Underdetermined blind speech separation with directivity pattern based continuous mask and ica," in *2004 12th European Signal Processing Conference*. IEEE, 2004, pp. 1991–1994.

[22] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *ICASSP*. IEEE, 2017, pp. 246–250.

[23] Y. Luo, Z. Chen, and N. Mesgarani, "Speaker-independent speech separation with deep attractor network," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 26, no. 4, pp. 787–796, 2018.

[24] J. R. Hershey, J. L. Roux, S. Watanabe, S. Wisdom, Z. Chen, and Y. Isik, "Novel deep architectures in speech processing," in *New Era for Robust Speech Recognition, Exploiting Deep Learning*, 2017, pp. 135–164. [Online]. Available: https://doi.org/10.1007/978-3-319-64680-0_6

[25] Z. Wang, J. L. Roux, and J. R. Hershey, "Alternative objective functions for deep clustering," in *ICASSP*. IEEE, 2018, pp. 686–690.

[26] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 241–245.

[27] M. Kolbaek, D. Yu, Z. Tan, and J. Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 25, no. 10, pp. 1901–1913, 2017.

[28] C. Xu, W. Rao, X. Xiao, E. S. Chng, and H. Li, "Single channel speech separation with constrained utterance level permutation invariant training using grid LSTM," in *ICASSP*. IEEE, 2018, pp. 6–10.

[29] L. Zhang, Z. Shi, J. Han, A. Shi, and D. Ma, "Furcanext: End-to-end monaural speech separation with dynamic gated dilated temporal convolutional networks," in *International Conference on Multimedia Modeling*. Springer, 2020, pp. 653–665.

[30] L. Wan, Q. Wang, A. Papir, and I. Lopez-Moreno, "Generalized end-to-end loss for speaker verification," in *ICASSP*. IEEE, 2018, pp. 4879–4883.

[31] N. Zeghidour, G. Synnaeve, N. Usunier, and E. Dupoux, "Joint learning of speaker and phonetic similarities with siamese networks," in *INTERSPEECH*. ISCA, 2016, pp. 1295–1299.

[32] M. Delcroix, K. Zmolikova, K. Kinoshita, A. Ogawa, and T. Nakatani, "Single channel target speaker extraction and recognition with speaker beam," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5554–5558.

[33] X. Xiao, Z. Chen, T. Yoshioka, H. Erdogan, C. Liu, D. Dimitriadis, J. Droppo, and Y. Gong, "Single-channel speech extraction using speaker inventory and attention network," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 86–90.

[34] C. Xu, W. Rao, E. S. Chng, and H. Li, "Time-domain speaker extraction network," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 327–334.

[35] T. Ochiai, M. Delcroix, K. Kinoshita, A. Ogawa, and T. Nakatani, "A unified framework for neural speech separation and extraction," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6975–6979.

[36] L. Drude, T. von Neumann, and R. Haeb-Umbach, "Deep attractor networks for speaker re-identification and blind source separation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 11–15.

[37] M. Delcroix, K. Zmolikova, T. Ochiai, K. Kinoshita, S. Araki, and T. Nakatani, "Compact network for speakerbeam target speaker extraction,"

in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6965–6969.

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

[39] T. von Neumann, K. Kinoshita, M. Delcroix, S. Araki, T. Nakatani, and R. Haeb-Umbach, “All-neural online source separation, counting, and diarization for meeting analysis,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*. IEEE, 2019, pp. 91–95. [Online]. Available: https://doi.org/10.1109/ICASSP.2019.8682572

[40] Y. Liu, M. Delfarah, and D. Wang, “Deep casa for talker-independent monaural speech separation,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[41] H. Chen and P. Zhang, “Exploring the time-domain deep attractor network with two-stream architectures in a reverberant environment,” *arXiv preprint arXiv:2007.00272*, 2020.

[42] J. Bernardes and D. Ayres-de Campos, “The persistent challenge of foetal heart rate monitoring,” *Current Opinion in Obstetrics and Gynecology*, vol. 22, no. 2, pp. 104–109, 2010.

[43] E. Karvounis, M. Tsipouras, C. Papaloukas, D. Tsalikakis, K. Naka, and D. Fotiadis, “A non-invasive methodology for fetal monitoring during pregnancy,” *Methods of information in medicine*, vol. 49, no. 03, pp. 238–253, 2010.

[44] I. Silva, J. Behar, R. Sameni, T. Zhu, J. Oster, G. Clifford, and G. Moody, “Noninvasive fetal ecg: The physionet/computing in cardiology challenge 2013,” *Computing in Cardiology*, pp. 149–152, 2013.

[45] P. Sutha and V. Jayanthi, “Fetal electrocardiogram extraction and analysis using adaptive noise cancellation and wavelet transformation techniques,” *Journal of medical systems*, vol. 42, no. 1, p. 21, 2018.

[46] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, “SDR - half-baked or well done?” in *ICASSP*. IEEE, 2019, pp. 626–630.

[47] E. Vincent, R. Gribonval, and C. Févotte, “Performance measurement in blind audio source separation,” *IEEE Trans. Audio, Speech & Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[48] Y. Linde, A. Buzo, and R. Gray, “An algorithm for vector quantizer design,” *IEEE Transactions on communications*, vol. 28, no. 1, 1980.

[49] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[50] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.

[51] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*.

[52] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, “Film: Visual reasoning with a general conditioning layer,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018, pp. 3942–3951.

[53] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” in *SSW*. ISCA, 2016.

[54] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. Le Roux, “Wham!: Extending speech separation to noisy environments,” in *Proc. Interspeech*, Sep. 2019, pp. 1368–1372.

[55] M. Maciejewski, G. Wichern, E. McQuinn, and J. L. Roux, “Whamr!: Noisy and reverberant single-channel speech separation,” *ArXiv*, vol. abs/1910.10279, 2019.

[56] I. Mccowan, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, D. Reidsma, and P. Wellner, “The ami meeting corpus,” in *In: Proceedings Measuring Behavior 2005, 5th International Conference on Methods and Techniques in Behavioral Research. L.P.J.J. Noldus, F. Grieco, L.W.S. Loijens and P.H. Zimmerman (Eds.), Wageningen: Noldus Information Technology*, 2005.

[57] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, “The fifth ’chime’ speech separation and recognition challenge: Dataset, task and baselines,” in *INTERSPEECH*. ISCA, 2018, pp. 1561–1565.

[58] S. T. Wisdom, H. Erdogan, D. Ellis, R. Serizel, N. Turpault, E. Fonseca, J. Salamon, P. Seetharaman, and J. Hershey, “What’s all the fuss about free universal sound separation data?” *ArXiv*, vol. abs/2011.00803, 2020.

[59] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2670313

[60] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. [Online]. Available: https://openreview.net/forum?id=r1Ddp1-Rb

[61] A. Sablayrolles, M. Douze, C. Schmid, and H. Jégou, “Spreading vectors for similarity search,” in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. [Online]. Available: https://openreview.net/forum?id=SkGuG2R5tm

[62] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *ICLR (Poster)*, 2015.

[63] S. Uhlich, M. Porcu, F. Giron, M. Enenkl, T. Kemp, N. Takahashi, and Y. Mitsufuji, “Improving music source separation based on deep neural networks through data augmentation and network blending,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 261–265.

[64] A. Défossez, N. Usunier, L. Bottou, and F. Bach, “Demucs: Deep extractor for music sources with extra unlabeled data remixed,” *ArXiv*, vol. abs/1909.01174, 2019.

[65] [Online]. Available: http://www.merl.com/demos/deep-clustering

[66] J. S. Garofolo, D. Graff, D. Paul, and D. S. Pallett, “CSR-I (WSJ0) complete,” Linguistic Data Consortium, Tech. Rep., 1993.

[67] J. Cosentino, M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, “Librimix: An open-source dataset for generalizable speech separation,” *arXiv preprint arXiv:2005.11262*, 2020.

[68] [Online]. Available: http://craffel.github.io/mir_eval/

[69] Y. Isik, J. L. Roux, Z. Chen, S. Watanabe, and J. R. Hershey, “Single-channel multi-speaker separation using deep clustering,” in *INTERSPEECH*. ISCA, 2016, pp. 545–549.

[70] C. Li, L. Zhu, S. Xu, P. Gao, and B. Xu, “Cbldnn-based speaker-independent speech separation via generative adversarial training,” in *ICASSP*. IEEE, 2018, pp. 711–715.

[71] Y. Luo and N. Mesgarani, “Tasnet: Time-domain audio separation network for real-time, single-channel speech separation,” in *ICASSP*. IEEE, 2018, pp. 696–700.

[72] M. W. Y. Lam, J. Wang, D. Su, and D. Yu, “Mixup-breakdown: a consistency training method for improving generalization of speech separation models,” *CoRR*, vol. abs/1910.13253, 2019.

[73] Y. Liu and D. Wang, “Divide and conquer: A deep casa approach to talker-independent monaural speaker separation,” *arXiv preprint arXiv:1904.11148*, 2019.

[74] [Online]. Available: https://git.io/JvOT6

[75] [Online]. Available: https://git.io/JvOTV

[76] M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, “Filterbank design for end-to-end speech separation,” *ArXiv*, vol. abs/1910.10400, 2019.

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

[78] F. Andreotti, J. Behar, S. Zaunseder, J. Oster, and G. D. Clifford, “An open-source framework for stress-testing non-invasive foetal ecg extraction algorithms,” *Physiological measurement*, vol. 37, no. 5, 2016.

Neil Zeghidour is a research scientist at Google Brain. He previously graduated with a PhD in Machine Learning from École Normale Supérieure-PSL in Paris, jointly with Facebook AI Research. His main research interest is to integrate signal processing with deep learning to design fully learnable classification, processing and synthesis systems of audio.

David Grangier is a research scientist at Google Brain. David earned his PhD in 2008 from EPFL, Switzerland, advised by Samy Bengio. Since then, he worked at different industrial research labs, including Microsoft Research and Facebook AI Research. His work focuses on machine learning models for structured prediction problems. This includes ranking models for information retrieval and sequential models for natural language processing. He notably contributed novel neural architectures for image search, language modeling and machine translation.