

---

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

---

Neil Zeghidour<sup>1</sup> David Grangier<sup>1</sup>

## Abstract

We introduce Wavesplit, an end-to-end speech separation system. From a single recording of mixed speech, the model infers and clusters representations of each speaker and then estimates each source signal conditioned on the inferred representations. The model is trained on the raw waveform to jointly perform the two tasks. Our model infers a set of speaker representations through clustering, which addresses the fundamental permutation problem of speech separation. Moreover, the sequence-wide speaker representations provide a more robust separation of long, challenging sequences, compared to previous approaches. We show that Wavesplit outperforms the previous state-of-the-art on clean mixtures of 2 or 3 speakers (WSJ0-2mix, WSJ0-3mix), as well as in noisy (WHAM!) and reverberated (WHAMR!) conditions. As an additional contribution, we further improve our model by introducing online data augmentation for separation.

## 1. Introduction

Automatic speech separation aims at isolating individual speaker voices from a recording with overlapping speech. Application-wise, separation is particularly important in meeting or social event recordings. Separation can be used to improve the intelligibility of speech both for human listeners and speech recognition systems. Although prior work in source separation spans several decades (Vincent et al., 2018), it is still an active area of research (Hershey et al., 2016; Luo & Mesgarani, 2019; Luo et al., 2019). In particular, the recent years have seen the emergence of supervised deep neural networks for speaker separation. Two main families of algorithms have been proposed: frequency masking approaches (Hershey et al., 2016; Le Roux et al., 2019a) and time-domain approaches (Luo & Mesgarani, 2019; Luo et al., 2019). Frequency masking approaches output a time-

frequency mask for each identified speaker which can be applied to an input spectrogram to reconstruct each speaker’s recording. Time-domain approaches directly predict the output audio for each speaker. Our model, Wavesplit, is a time-domain approach, which bypasses the difficulty of phase reconstruction required by time-frequency approaches (Le Roux et al., 2019a).

Our work targets open speaker separation, i.e. the test speakers have not been encountered at training time, but draws inspiration from the closed speaker setting in order to characterize desirable properties for latent speaker representations. Our work considers a joint training procedure for speaker identification and speech separation, which differs from prior research (Wang et al., 2018a). Our training objective encourages identifying instantaneous speaker representations such that (i) these representations can be grouped into individual speaker clusters and (ii) the centroids of such clusters provide a long-term speaker representation for the reconstruction of individual speaker signals.

As an additional contribution orthogonal to our model, we highlight the benefits of the dynamic creation of audio mixtures online during training. Rather than training a model on a pre-computed, finite set of mixtures, we constantly create new training examples by mixing short windows of clean signal uniformly sampled from the training set, reweighted by randomly sampled gains. This strategy brings systematic and substantial improvements in the performance of the final system, and can easily be derived from any speech dataset.

Wavesplit improves the state-of-the-art over standard separation benchmarks, both for clean (WSJ0-2mix, WSJ0-3mix) and noisy datasets (WHAM! and WHAMR!). We also highlight the accuracy of our model when the number of speakers varies and when the relative loudness between speakers is variable. Augmentation by dynamic mixing further improves these results.

The contributions of this paper span different axes, (i) we leverage training speaker labels but do not require any information about the test speakers beside the mixture recording, (ii) we aggregate information about speakers over the whole sequence which is advantageous for long recordings, (iii) we rely on clustering for inferring speaker identity which naturally outputs sets, i.e. order-agnostic predictions, (iv) we report state-of-the-art results on the most common speech

---

<sup>1</sup>Google Research, Brain Team. Correspondence to: Neil Zeghidour <neilz-at-google-dot-com>, David Grangier <grangier-at-google-dot-com>.

separation benchmarks, in various conditions, (v) we analyze the empirical advantages and drawbacks of our method.

## 2. Related Work

Single-channel speech separation takes a single microphone recording in which multiple speakers overlap and predicts the isolated audio of each speaker. This classical signal processing task (Roweis, 2001; Yilmaz & Rickard, 2004; Vincent et al., 2018) has witnessed fast progress in the recent years thanks to supervised neural networks (Wang & Chen, 2018). Neural models can be mainly grouped in two families: spectrogram-based and time-domain approaches.

**Spectrogram-based approaches** rely on a time-frequency representation obtained through short-term Fourier Transform (Williamson, 2012). Considering the input mixture spectrogram, these approaches aim to assign each time-frequency bin (TFB) to one of the sources. For that purpose, the model produces one mask per source that identifies the source with maximal energy for each TFB. Source spectrograms are estimated by multiplying the input spectrogram with the corresponding masks. Source signals are then estimated by phase reconstruction (Griffin & Jae Lim, 1984; Wang et al., 2018c). Variants of this strategy include models with soft masking, i.e. assigning TFBs to multiple speakers with different weights (Araki et al., 2004). Hershey et al. (2016) devise a neural model capable of predicting multiple masks. They propose to learn a latent representation for each TFB mask such that the distance between TFB assigned to the same source is lower than the distance between TFB from different sources. At test time, clustering these representations allows identifying which TFBs belong to each source. Alternatively, (Yu et al., 2017; Kolbaek et al., 2017; Xu et al., 2018) avoid TFB clustering by predicting multiple masks directly. The predicted masks are compared to the target masks by searching over the permutations of the source ordering. This is called Permutation Invariant Training (PIT). PIT acknowledges that the order of predictions and labels in speech separation is irrelevant, i.e. separation is a set prediction problem.

**Time domain approaches** avoid the degradation introduced by phase reconstruction and the latency introduced by high resolution spectrograms. Time-domain models directly predict a set of audio outputs which are compared to the audio sources by searching over the permutations of the source ordering, i.e. PIT applied to time domain signal matching. Multiple architectures have been proposed for this scheme, including convolutional (Luo & Mesgarani, 2019; Zhang et al., 2020) and recurrent networks (Luo et al., 2019).

**Leveraging speaker representation** for speech separation is another axis of research relevant to our work. Short recordings of speech are sufficient to extract discriminant

speaker characteristics (Wan et al., 2018; Zeghidour et al., 2016). This property has been used for speech separation in the past. For instance, Wang et al. (2018a) extract the characteristics of a speaker from a clean enrollment sequence in order to subsequently isolate this speaker in a mixture. This method however does not apply to open-set speaker separation, where no information about the test speakers is available besides the input mixture.

## 3. Wavesplit

Our strategy relies on a convolutional architecture composed of a speaker stack and a separation stack, see Figure 1. The speaker stack takes the input mixture and produces a set of vectors representing the speakers present in the mixture. The separation stack consumes both the input mixture and the set of speaker representations from the speaker stack. It produces a multi-channel audio output with separated speech from each speaker in the set.

The separation stack is classical and resembles previous architectures conditioned on pre-trained speaker vectors (Wang et al., 2018a), or trained with PIT (Luo & Mesgarani, 2019). The speaker stack is novel and constitutes the heart of our contribution. This stack is trained jointly with the separation stack. At training time, it relies on speaker labels 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. In contrast with VoiceFilter (Wang et al., 2018a), the representation of all speakers is directly inferred from the mixture and we do not need an enrollment sequence for test speakers. With joint training, the speaker representation is not solely optimized for identification but also extracts information necessary for isolated speech reconstruction. In contrast with PIT (Yu et al., 2017), 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 4.3. 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.

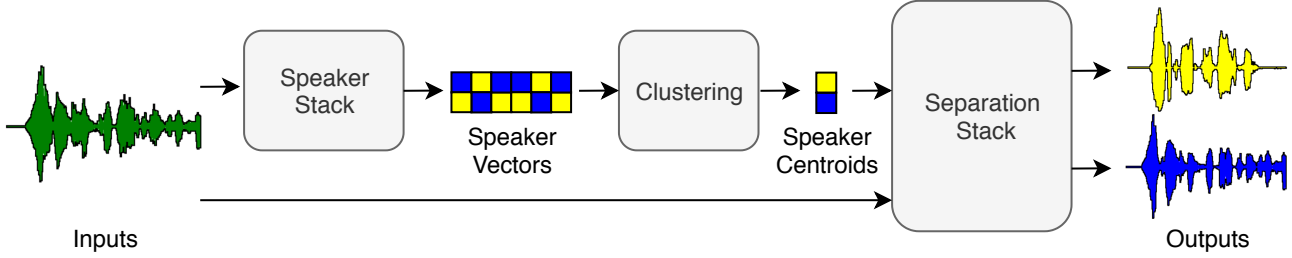


Figure 1. Wavesplit for 2-speaker separation. The speaker stack extracts speaker vectors at each timestep. The vectors are clustered and aggregated in speaker centroids. The separation stack ingests the centroids and the input signal to output two clean channels.

### 3.1. 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 (in samples). The task of source separation is to reconstruct each  $y^i$  from an input  $x$ , which sums all the sources,  $x = \sum_{i=1}^N y^i$ .

A separation model  $f$  predicts an estimate for each channel  $i = 1, \dots, N$ ,

$$\hat{y}^i = f^i(x).$$

These estimates should match the reference recordings  $\{y^i\}_{i=1}^N$  up to a permutation since the channel ordering is arbitrary. The quality of the reconstruction is usually computed as,

$$Q(\hat{y}, y) = \max_{\sigma \in S_N} \frac{1}{N} \sum_{i=1}^N q(\hat{y}^{\sigma(i)}, y^i) \quad (1)$$

where  $q(\cdot, \cdot)$  denotes a single-channel reconstruction quality metric and  $S_N$  denotes the space of permutation over  $[1, N]$ . The speaker separation literature typically relies on Signal-to-Distortion Ratio (SDR) to assess reconstruction quality,

$$\text{SDR}(\hat{y}, y) = -10 \log_{10} \left( \frac{\|y - \hat{y}\|^2}{\|y\|^2} \right)$$

i.e. the opposite of the log squared error normalized by the energy of the reference signal. Scale-invariant SDR (SI-SDR) (Le Roux et al., 2019b) searches over rescalings of the signal to not penalize models for a wrong scale of predictions. Variants searching over richer signal transforms have also been proposed (Vincent et al., 2006).

### 3.2. Model Architecture

Wavesplit is a residual convolutional network with two sub-networks 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.

$$\{h^i\}_{i=1}^N = h(x).$$

where  $\forall i, h^i \in \mathbb{R}^{T \times d}$ .  $N$  is chosen to upper bound 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. When  $n < N$  speakers are present at a time-step,  $h$  produces a latent representation of silence for the  $N - n$  remaining outputs. 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 produced vectors by speaker and outputs  $N$  summary vectors for the whole sequence. K-means performs this aggregation at inference (Linde et al., 1980) and returns the centroids of the  $N$  identified clusters,

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

During training, the aggregation is derived from a speaker training objective described in Section 3.3. 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*.

The **separation stack** takes the vectors  $c = \{c_i\}_{i=1}^N$  and the input signal  $x$  and produces an  $N$ -channel output  $\hat{y}$ ,

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

We rely on a **residual convolutional architecture** similar to Luo & Mesgarani (2019) for both stacks. Our residual block (He et al., 2016) for the speaker stack composes a dilated convolution  $\text{dconv}$  (Yu & Koltun, 2016), a non-linearity  $\text{nl}$  and layer normalization  $\text{lnorm}$  (Ba et al., 2016),

$$x_{l+1} = x_l + \text{lnorm}(\text{nl}(\text{dconv}(x_l))). \quad (2)$$

We experimented with different non-linearities including rectified linear units (Glorot et al., 2011), parametric rectified linear units (He et al.), gated linear units and gated

tangent units (Dauphin et al., 2017). Parametric rectified linear units were consistently better on our validation experiments and were selected for our subsequent experiments. The last layer of the speaker stack applies Euclidean normalization to the speaker vectors.

The conditioning of the separation stack by the speaker centroids relies on FiLM, Feature-wise Linear Modulation (Perez et al., 2018). At each layer, we apply a linear transformation  $\text{lin}$  to the speaker centroids, and add FiLM conditioning on a block similar to Eq. (2),

$$x_{l+1} = x_l + \text{Inorm}(\text{nl}(a * \text{dconv}(x) + b))$$

where  $a = \text{lin}(c)$  and  $b = \text{lin}'(c)$  are different linear projections of  $c$ , the concatenation of the speaker centroids. We learn distinct parameters for each layer for all parametric functions, i.e.  $\text{dconv}$ ,  $\text{nl}$ ,  $\text{Inorm}$ ,  $\text{lin}$  and  $\text{lin}'$ . Section 4.6 shows the advantage of FiLM conditioning on generalization and convergence compared to the bias only conditioning ( $a = 1$ ) used commonly (van den Oord et al., 2016).

### 3.3. Model Training Objective

Model training follows a dual objective: (i) it learns speaker vectors which can be clustered by 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. This information is provided by most speaker separation datasets, including Wall Street Journal variants (Hershey et al., 2016; Wichern et al., 2019; Maciejewski et al., 2019), meeting recordings (Mccowan et al., 2005) or cocktail party recordings (Barker et al., 2018). We experiment with different loss functions to encourage the speaker stack outputs to have small intra-speaker and large inter-speaker distances.

The **speaker vector objective** leverages training speaker identities. Given an input sequence  $x$  with target signals  $\{y^i\}_{i=1}^N$  and corresponding speakers  $\{s_i\}_{i=1}^N \in [1, M]^N$ , we consider different loss functions which encourage the correct identification of the  $N$  speakers 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  $\ell_{\text{speaker}}$  defines a loss function between 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 (Hershey et al., 2017). This permutation per time-step differs from PIT (Xu et al., 2018): we do not force the model to pick a single ordering over the output channels and we train the separation stack with different permutations of the same labels.

For  $\ell_{\text{speaker}}$ , we explore three alternatives. In all cases, we maintain an embedding table over training speakers  $E \in \mathbb{R}^{M \times d}$ . First, we consider a distance objective  $\ell_{\text{speaker}}^{\text{dist}}$ . This loss favors small distances between a speaker vector and the corresponding embedding. It also enforces the distance between speaker vectors at the same time-step to be larger than a margin of 1, i.e.

$$\ell_{\text{speaker}}^{\text{dist}}(h_t^j, s_i) = \|h_t^j - E_{s_i}\|^2 + \sum_{k \neq j} \max(0, 1 - \|h_t^j - h_t^k\|^2) \quad (3)$$

Second, we consider a local classifier objective  $\ell_{\text{speaker}}^{\text{local}}$ . This classifier discriminates among the speakers present in the sequence. It relies on the log softmax function over the distance between speaker vectors and speaker embeddings, i.e.

$$\ell_{\text{speaker}}^{\text{local}}(h_t^j, s_i) = -d(h_t^j, E_{s_i}) - \log \sum_{k=1}^N \exp(-d(h_t^j, E_{s_k})) \quad (4)$$

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.

$$\ell_{\text{speaker}}^{\text{global}}(h_t^j, s_i) = -d(h_t^j, E_{s_i}) - \log \sum_{k=1}^M \exp(-d(h_t^j, E_k)) \quad (5)$$

The **reconstruction objective** aims at optimizing the separation quality, as defined in Eq. (1).

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

Contrasting with Permutation Invariant Training, PIT (Yu et al., 2017), this expression does not require searching over the space of permutation since the vectors  $c = \{c_i\}_{i=1}^N$  is consistent with the order of the labels  $\{y^i\}_{i=1}^N$  as explained above.

For  $\ell_{\text{reconstr}}$ , we use negative SDR with a clipping threshold  $\tau$  to limit the influence of the high quality training predictions,

$$\ell_{\text{reconstr}}(\hat{y}, y) = -\min(\tau, \text{SDR}(\hat{y}, y)).$$

In addition to these two losses, we consider two forms of regularization to reduce the mismatch between training and testing conditions. First, to improve generalization to new speakers, we favor well separated embeddings for the speakers of the training set. We achieve this by adding the entropy regularization from (Sablayrolles et al., 2019),

$$\ell_{\text{reg}} = - \sum_i \min_{j \neq i} \log \|E_i - E_j\|.$$

Second, to make the separation stack robust to noisy speaker vectors, we add Gaussian noise on these vectors during training. Section 4.6 reports an ablation that demonstrates the benefits of regularization.

### 3.4. Training Algorithm

Model training optimizes the weighted sum of  $\mathcal{L}_{\text{speaker}}$  and  $\mathcal{L}_{\text{reconstr}}$  with the Adam algorithm (Kingma & Ba, 2015). We train on mini batches of fixed-size windows. We found that the choice of window size is not critical for Wavesplit, even when being as small as 750ms, unlike most PIT-based approaches (Luo & Mesgarani, 2019; Luo et al., 2019) that require longer segments ( $\sim 4$ s). The training set is shuffled at each epoch and a window starting point is uniformly sampled each time a sequence is visited. This sampling means that each training sequence is given the same importance regardless of its length. This strategy is consistent with the averaging of per-sequence SDR commonly used for evaluation. To prevent a potential over-fitting to the ordering of the training label, we replicated the training data for all permutations of the source signals. Replication, windowing and shuffling are not applied to the validation or test set.

### 3.5. Data Augmentation with Dynamic Mixing

Source separation aims at isolating individual voices in a signal summing different sources. Source separation benchmarks such as WSJ0-2mix (Hershey et al., 2016) create a standard split between train, valid and test sequences and then generate specific input mixtures by sampling pairs of sequences along with corresponding gains to apply prior to summation. As an orthogonal contribution to the Wavesplit model, we evaluate the impact of dynamically creating mixtures during training. Our augmentation algorithm receives a batch of  $B \times N$  targets windows and creates  $B$  groups of size  $N$  at random by shuffling the data. We then sample gains for each of the  $B$  mixtures and then sum the  $N$  sequences to obtain each artificial input. In other words, we replicate the creation recipe of the dataset indefinitely. Despite its simplicity, our experiments report a strong improvement from this method. A similar mixing augmentation scheme has been used in music source separation (Uhlich et al., 2017; Défossez et al., 2019). Our experiments also report results without augmentation to isolate the impact of Wavesplit alone.

## 4. Experiments & Results

Most of our experiments are performed on the speaker separation dataset<sup>1</sup> built from the LDC WSJ-0 dataset (Garofolo et al., 1993) as introduced in (Hershey et al., 2016). We rely on the 8kHz version of the dataset, with 2 or 3 concurrent speakers. This setting has become the de-facto benchmark for open-speaker source separation and we compare our results to alternative methods. Table 1 reports dataset statistics. We also introduce a version of the dataset in which the number of active speakers varies from one to three within a sequence as a more realistic setting. Additionally, we perform experiments on the noised versions of the data. We rely on WHAM! with urban noise (Wichern et al., 2019) and WHAMR! with noise and reverberation (Maciejewski et al., 2019). These datasets are derived from WSJ0-2mix and have identical statistics. We further compare different variations of Wavesplit varying the optimized loss functions and architectural alternatives. Finally, we also conduct an error analysis examining a small fraction of sequences which have a strong negative impact on overall performance.

Table 1. WSJ0-2/3mix statistics.

Dataset		train	valid	test
WSJ0-2mix	# examples	20k	5k	3k
	# speakers	101		
	mean length	5.4 sec	5.5 sec	5.7 sec
WSJ0-3mix	# sequences	20k	5k	3k
	# speakers	101		
	mean length	4.9 sec	4.9 sec	5.2 sec

Our evaluation relies on signal-to-distortion ratio (SDR) and scale-invariant signal-to-distortion ratio (SI-SDR) (Vincent et al., 2006; Le Roux et al., 2019b), as introduced in Section 3.1. SDR is measured using the standard MIR eval library<sup>2</sup>. Consistently with the literature, we measure both metrics in terms of *improvement*, i.e. the metric obtained using the system output minus the metric obtained by using the input mixture as the prediction.

### 4.1. Hyperparameter Selection

The choice of architecture is based on preliminary validation experiments on the 2 speaker separation task of WSJ0-2mix (Hershey et al., 2016), and then used through all our experiments. We selected a latent dimension of 512 for both stacks. For the dilated convolutions, we selected a kernel of size 3 without striding, therefore all activations have the same temporal resolution as the input signal and no up-sampling is necessary to produce the output signal. The dilation factor varies with depth. The speaker stack is 14-

<sup>1</sup><http://www.merl.com/demos/deep-clustering>

<sup>2</sup>[http://craffel.github.io/mir\\_eval/](http://craffel.github.io/mir_eval/)

layer deep with dilation growing exponentially from  $2^0$  to  $2^{13}$ . The separation stack has 40 layers with a dilation pattern borrowed from Wavenet (van den Oord et al., 2016), i.e.  $\delta_l = 2^{l \bmod 10}$ . Every 10 layer the dilation is reset to 1 allowing multiple fine-to-coarse-to-fine interactions across the time axis.

After architecture selection, we experimented with learning rates in  $[1e-3, 2e-3, 3e-3]$  and speaker loss weights in  $[5, 10, 15]$ . We found that  $2e-3$  and 10 were respectively the best hyperparameters and thus used the same in our other experiments. For regularization, we trained models with a distance regularization weight in  $[0, 0.2, 0.3, 0.5]$  and a Gaussian noise with standard deviation in  $[0, 0.1, 0.2, 0.3]$ . We found 0.3 and 0.2 to be the best respectively. The clipping threshold on the negative SDR loss was picked between  $[22, 27, 30]$  for each dataset. For clean data, 30 was the best, while 27 was superior in noisy conditions.

For  $\mathcal{L}_{\text{speaker}}$  alternatives, we found the global classifier to be the most effective, see Section 4.6.

## 4.2. Clean Settings

WSJ0-2mix/3-mix is the de facto benchmark for separation. Table 2 reports the results for two simultaneous speakers, while Table 4 reports the results for three simultaneous speakers. In both cases, Wavesplit outperforms alternatives and dynamic mixing further increases this advantage. For instance, we report 19.2  $\Delta$ SDR compared to 19.0 for the recent dual path RNN (Luo et al., 2019). This number improves to 20.6 with dynamic augmentation. We also made available recordings processed by our system on an anonymous webpage<sup>3</sup>, as well as in supplementary material.

We examine further the accuracy on WSJ0-2mix. Table 3 reports our analysis on the error distribution. Our results show that the test set exhibits a larger fraction of sentences with  $< 5$  dB  $\Delta$ SDR than the validation set, i.e. 7.2% versus 1.3%. The WSJ0-2mix validation set contains the same speakers as the training data, we therefore examine further if the lower test performance is due to the representation of the test speakers. We rely on the training speaker classifier from Eq. 5 and collect the posterior distribution over training speakers for each speaker vector and average it per cluster. When the same training identity collects more than 10% of the probability mass in both clusters, we label the sequence as "confusing speakers" since both clusters contains vectors confused with the same training identity. We observe that low SDR correlates with situations where both speakers in the test recordings are mapped to the same training identity. The training set of WSJ0-2mix has only 101 speakers. We suspect that training on datasets with a

wider range of speakers will alleviate this problem and will be studied in future work.

Additionally, we evaluate oracle  $\Delta$ SDR. This metric reports  $\Delta$ SDR when the predictions at each time step are permuted to best match the labels. The 7.2% of test examples with poor performance reach 16.6 oracle  $\Delta$ SDR. This means that poor performing predictions are mostly the result of channel permutation throughout the sequence, i.e. the model reconstructs the signal properly but does not have a consistent speaker assignment between channels. This channel permutation is problematic for speaker separation applications and it will be particularly beneficial to focus on low  $\Delta$ SDR quantiles specifically in future research.

## 4.3. Robustness to Changes in Dominant Speaker

The examples in WSJ0-2mix blend two speakers without changes in the dominant speaker, i.e. the same speaker stays the loudest throughout the whole recording. To evaluate our model in more challenging conditions, we create longer sequences by concatenating WSJ0-2mix examples for the same pair of speakers. We pick the concatenated recordings such that the dominating speaker alternates between the two identities. In this setting, the speaker with the highest energy varies along time. This experiment stems from research prior to PIT (Weng et al., 2015) which suggested that average energy was a good rule for channel assignment, applicable when one channel consistently dominates the other. We suspect that PIT may implicitly rely on energy to address the channel assignment ambiguity. In this case, the change in dominating speaker may degrade accuracy for such a model and even lead to cross-channel contamination, i.e. some segments of speech are swapped across speaker channels in the prediction. Wavesplit is agnostic to loudness dependent ordering since the separation stack is trained from all permutations of speaker identities at training time.

Our comparison relies on models trained on the regular WSJ0-2mix training set and only changes test conditions. We vary the sequence length from 1 to 10 times the original length. For PIT models, we retrained Conv-TasNet with an open source implementation<sup>4</sup>, and reproduced the 15.6  $\Delta$ SDR improvement of (Luo & Mesgarani, 2019). For Dual-Path RNN we used a pre-trained model<sup>5</sup> which gives 19.1  $\Delta$ SDR (against 19.0 in (Luo et al., 2019)). These models are among the best prior art on this dataset, yet they degrade significantly faster with longer sequences and alternating dominant speakers compared to our model, as shown in Table 5. This is particularly remarkable as Wavesplit is trained on only 1s long windows, unlike ConvTasNet and Dual-Path RNN that are both trained on 4s long windows.

<sup>3</sup><https://soundcloud.com/wavesplitdemo/sets>

<sup>4</sup><https://git.io/JvOT6>

<sup>5</sup><https://git.io/JvOTV>

Table 2. SI-SDR and SDR improvements (dB) on WSJ0-2mix.

Model	$\Delta$ SI-SDR	$\Delta$ SDR
Deep Clustering (Isik et al., 2016)	10.8	–
uPIT-blstm-st (Kolbaek et al., 2017)	–	10.0
Deep Attractor Net. (Chen et al., 2017)	10.5	–
Anchored Deep Attr. (Luo et al., 2018)	10.4	10.8
Grid LSTM PIT (Xu et al., 2018)	–	10.2
ConvLSTM-GAT (Li et al., 2018)	–	11.0
Chimera++ (Wang et al., 2018b)	11.5	12.0
WA-MISI-5 (Wang et al., 2018c)	12.6	13.1
blstm-TasNet (Luo & Mesgarani, 2018)	13.2	13.6
Conv-TasNet (Luo & Mesgarani, 2019)	15.3	15.6
Conv-TasNet+MBT (Lam et al., 2019)	15.5	15.9
DeepCASA (Liu & Wang, 2019)	17.7	18.0
FurcaNeXt (Zhang et al., 2020)	–	18.4
DualPathRNN (Luo et al., 2019)	18.8	19.0
Wavesplit	<b>19.0</b>	<b>19.2</b>
Wavesplit + Dynamic mixing	<b>20.4</b>	<b>20.6</b>

Table 3. Error Analysis on WSJ0-2mix.

Set	Metric	$\Delta$ SDR (dB)	
		< 5.0	$\geq$ 5.0
Valid	% Examples	1.3	98.7
	% Confusing speakers	69.2	4.8
	mean $\Delta$ SDR	0.40	22.1
	oracle $\Delta$ SDR	15.6	22.7
Test	% Examples	7.2	92.8
	% Confusing speakers	56.0	12.5
	mean $\Delta$ SDR	0.51	20.7
	oracle $\Delta$ SDR	16.6	22.0

#### 4.4. Variable Number of Active Speakers

The evaluation protocol in the previous section considers a fixed given number of active speakers. The model is trained on recordings with the same number of speakers throughout the whole recording and tested in the same conditions. In this section, we train and test our model over sequences with at most 1, 2 or 3 speakers active at a time. We create this dataset WSJ0-123mix by padding WSJ0-3mix with silence. Each sequence in WSJ0-3mix is padded three times with the padding patterns shown in Figure 2.

Compared to previous settings, target windows with silence or partial silence induce large variations in the ground-truth signal norm and yield numerical instabilities when learning with the SDR loss. We therefore replace the normalization by the reference energy with a constant energy corresponding to the average norm of the reference training windows.

Table 4. SI-SDR and SDR improvements (dB) on WSJ0-3mix.

Model	$\Delta$ SI-SDR	$\Delta$ SDR
Deep Clustering (Isik et al., 2016)	7.1	–
uPIT-blstm-st (Kolbaek et al., 2017)	–	7.7
Deep Attractor Net. (Chen et al., 2017)	8.6	8.9
Anchored Deep Attr. (Luo et al., 2018)	9.1	9.4
Conv-TasNet (Luo & Mesgarani, 2019)	12.7	13.1
Wavesplit	<b>15.4</b>	<b>15.8</b>
Wavesplit + Dynamic mixing	<b>16.5</b>	<b>16.8</b>

Table 5. SDR improvement (dB) on longer test sequences with alternating dominant speaker (custom WSJ0-2mix test sets).

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	<b>19.2</b>	<b>18.8</b>	<b>18.3</b>

Table 8 reports a breakdown of our results. The sequences with no speech overlap are excellent, highlighting that the model correctly identifies speaker turns and isolates speakers. This suggests that our model could address diarization (Anguera et al., 2012). Conversely, the sequences with three continuously active speakers report an accuracy lower than our model trained on WSJ0-3mix. We suspect that our loss clipping strategy is sub-optimal on this dataset. To address this issue, we plan to explore losses and training schemes that emphasize low performing quantiles during training in future work (Kibzun & Kurbakovskiy, 1991).

#### 4.5. Noisy and Reverberated Settings

WSJ0-2mix was recorded in clean and controlled conditions and noisy variants have been introduced to represent more challenging use cases. WHAM! (Wichern et al., 2019) adds noise recorded in public areas to the mixtures. As the model should only predict clean signals, it cannot exploit the fact that predicted channels should sum to the input signal. WHAMR! (Maciejewski et al., 2019) adds the same noise, but also reverberates the clean signals. This makes the task even harder, as the model has to predict anechoic clean signals (i.e. without reverberation), and therefore solve jointly the tasks of denoising, dereverberation and source separation. Tables 6 and 7 show that our models outperform previous work by a substantial margin, even without dynamic mixing. We also 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!,

Table 6. SI-SDR and SDR improvements (dB) on WHAM!.

Model	$\Delta$ SI-SDR	$\Delta$ SDR
Chimera++ (Wichern et al., 2019)	9.9	–
Conv-TasNet (Pariante et al., 2019)	12.7	–
Learnable fbank (Pariante et al., 2019)	12.9	–
Wavesplit	<b>14.5</b>	<b>15.0</b>
Wavesplit + Dynamic mixing	<b>15.3</b>	<b>15.8</b>

Table 7. SI-SDR and SDR improvements (dB) on WHAMR!.

Model	$\Delta$ SI-SDR	$\Delta$ SDR
Conv-TasNet (Maciejewski et al., 2019)	8.3	–
BLSTM-TasNet (Maciejewski et al., 2019)	9.2	–
Wavesplit	<b>11.3</b>	<b>10.3</b>
Wavesplit + Dynamic mixing	<b>13.0</b>	<b>11.9</b>

except that we reweighted reverberated signals with noise, similarly to the original training set. On both datasets, this leads to an even larger improvement over previous work. For instance, the results on WHAMR! are comparable to the reconstruction accuracy from clean inputs (WSJ0-2mix) with models prior to (Wang et al., 2018b).

#### 4.6. Ablation Study

Table 9 compares the base result obtained with the global classifier loss, Eq. (5), with the distance loss, Eq. (3). Although this type of loss is common in distance learning for clustering (Wang et al., 2018b), the global classifier reports better results. We also ran experiments with the local classifier loss, Eq. (4), which was found to yield very slow training and worse generalization.

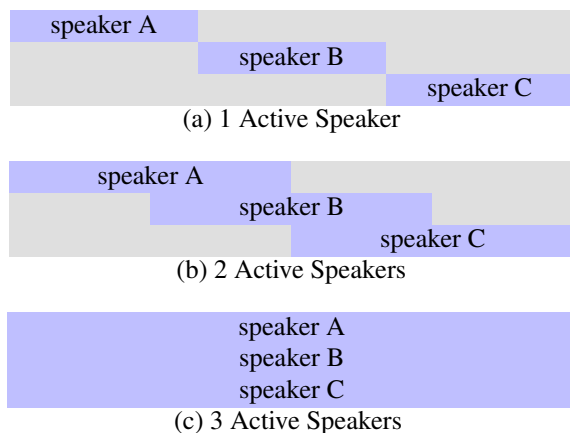


Figure 2. Speaker overlap patterns for the target channels for WSJ0-123mix. Gray refers to silence padding.

Table 8. Wavesplit performance when the number of speakers varies (WSJ0-123mix, no dynamic mixing).

Subset	$\Delta$ SI-SDR (dB)	$\Delta$ SDR (dB)
1 active speaker	18.6	19.2
2 active speakers	15.7	16.3
3 active speakers	13.3	13.7
Overall	15.9	16.4

Table 9. Independent impact of conditioning &amp; regularization alternatives on WSJ0-2mix (validation set, no dynamic mixing).

Model	$\Delta$ SDR (dB)
Base model	21.3
w/ distance loss	20.8
w/o FiLM	20.7
w/o distance reg.	21.0
w/o gaussian noise	20.9

Table 9 also reports the advantage of multiplicative FiLM conditioning compared to more classical additive conditioning (van den Oord et al., 2016). Not only reported SDR are better but FiLM allows using a higher learning rate and hence enables faster training. Table 9 also shows the benefit of regularization of the speaker representation.

## 5. Conclusions

We introduce Wavesplit, a neural network for speech separation. Our model decomposes the source separation problem in two jointly trained tasks, i.e. our model extracts a representation for each speaker present in the recording and performs separation conditioned on the inferred representation. Inference therefore relies on a consistent representation of each speaker throughout the sequence. This allows the model to explicitly keep track of speaker/channel assignments, limiting channel swapping, i.e. inconsistencies in such assignments.

Our approach is advantageous compared to prior work and redefines the state-of-the-art on standard benchmarks, both in clean and noisy conditions. We also observe the advantage of our model when the number of speaker varies throughout the sequence and when the relative loudness of speakers varies.

We want to extend this work by introducing an autoregressive or flow-based decoder to further enhance speech outputs, especially for noisy inputs for which the separation problem is particularly under-determined. We also see potential practical benefits in increasing the influence of low accuracy examples during training, e.g. by optimizing directly loss quantiles.



## 6. Acknowledgments

The authors are grateful to Adam Roberts, Chenjie Gu and Raphael Marinier for their advice on model implementation. They are also grateful to Jonathan Le Roux, John Hershey, Richard F. Lyon and Norman Casagrande for their help navigating speech separation prior work.

## References

- Anguera, X., Bozonnet, S., Evans, N., Fredouille, C., Friedland, G., and Vinyals, O. Speaker diarization: A review of recent research. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):356–370, 2012.
- Araki, S., Makino, S., Sawada, H., and Mukai, R. Underdetermined blind speech separation with directivity pattern based continuous mask and ica. In *2004 12th European Signal Processing Conference*, pp. 1991–1994. IEEE, 2004.
- Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- Barker, J., Watanabe, S., Vincent, E., and Trmal, J. The fifth ‘chime’ speech separation and recognition challenge: Dataset, task and baselines. In *INTERSPEECH*, pp. 1561–1565. ISCA, 2018.
- Chen, Z., Luo, Y., and Mesgarani, N. Deep attractor network for single-microphone speaker separation. In *ICASSP*, pp. 246–250. IEEE, 2017.
- Dauphin, Y. N., Fan, A., Auli, M., and Grangier, D. Language modeling with gated convolutional networks. In *International Conference on Machine Learning (ICML)*, 2017.
- Défossez, A., Usunier, N., Bottou, L., and Bach, F. Demucs: Deep extractor for music sources with extra unlabeled data remixed. *ArXiv*, abs/1909.01174, 2019.
- Garofolo, J. S., Graff, D., Paul, D., and Pallett, D. S. CSR-I (WSJ0) complete. Technical report, Linguistic Data Consortium, 1993.
- Glorot, X., Bordes, A., and Bengio, Y. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, pp. 315–323, 2011.
- Griffin, D. and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1984.
- He, K., Zhang, X., Ren, S., and Sun, J. 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*.
- He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- Hershey, J. R., Chen, Z., Roux, J. L., and Watanabe, S. Deep clustering: Discriminative embeddings for segmentation and separation. In *ICASSP*, pp. 31–35. IEEE, 2016.
- Hershey, J. R., Roux, J. L., Watanabe, S., Wisdom, S., Chen, Z., and Isik, Y. Novel deep architectures in speech processing. In *New Era for Robust Speech Recognition, Exploiting Deep Learning*, pp. 135–164. 2017. doi: 10.1007/978-3-319-64680-0\_6. URL [https://doi.org/10.1007/978-3-319-64680-0\\_6](https://doi.org/10.1007/978-3-319-64680-0_6).
- Isik, Y., Roux, J. L., Chen, Z., Watanabe, S., and Hershey, J. R. Single-channel multi-speaker separation using deep clustering. In *INTERSPEECH*, pp. 545–549. ISCA, 2016.
- Kibzun, A. I. and Kurbakovskiy, V. Y. Guaranteeing approach to solving quantile optimization problems. *Annals of operations research*, 30(1):81–93, 1991.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.
- Kolbaek, M., Yu, D., Tan, Z., and Jensen, J. Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks. *IEEE/ACM Trans. Audio, Speech & Language Processing*, 25(10):1901–1913, 2017.
- Lam, M. W. Y., Wang, J., Su, D., and Yu, D. Mixup-breakdown: a consistency training method for improving generalization of speech separation models. *CoRR*, abs/1910.13253, 2019.
- Le Roux, J., Wichern, G., Watanabe, S., Sarroff, A. M., and Hershey, J. R. The phasebook: Building complex masks via discrete representations for source separation. In *ICASSP*, pp. 66–70. IEEE, 2019a.
- Le Roux, J., Wisdom, S., Erdogan, H., and Hershey, J. R. SDR - half-baked or well done? In *ICASSP*, pp. 626–630. IEEE, 2019b.
- Li, C., Zhu, L., Xu, S., Gao, P., and Xu, B. Cbldnn-based speaker-independent speech separation via generative adversarial training. In *ICASSP*, pp. 711–715. IEEE, 2018.
- Linde, Y., Buzo, A., and Gray, R. An algorithm for vector quantizer design. *IEEE Transactions on communications*, 28(1):84–95, 1980.

- Liu, Y. and Wang, D. Divide and conquer: A deep casa approach to talker-independent monaural speaker separation. *arXiv preprint arXiv:1904.11148*, 2019.
- Luo, Y. and Mesgarani, N. Tasnet: Time-domain audio separation network for real-time, single-channel speech separation. In *ICASSP*, pp. 696–700. IEEE, 2018.
- Luo, Y. and Mesgarani, N. Conv-tasnet: Surpassing ideal time-frequency magnitude masking for speech separation. *IEEE/ACM Trans. Audio, Speech & Language Processing*, 27(8):1256–1266, 2019.
- Luo, Y., Chen, Z., and Mesgarani, N. Speaker-independent speech separation with deep attractor network. *IEEE/ACM Trans. Audio, Speech & Language Processing*, 26(4):787–796, 2018.
- Luo, Y., Chen, Z., and Yoshioka, T. Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation. *arXiv preprint arXiv:1910.06379*, 2019.
- Maciejewski, M., Wichern, G., McQuinn, E., and Roux, J. L. Whamr!: Noisy and reverberant single-channel speech separation. *ArXiv*, abs/1910.10279, 2019.
- Mccowan, I., Lathoud, G., Lincoln, M., Lisowska, A., Post, W., Reidsma, D., and Wellner, P. 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.
- Pariente, M., Cornell, S., Deleforge, A., and Vincent, E. Filterbank design for end-to-end speech separation. *ArXiv*, abs/1910.10400, 2019.
- Perez, E., Strub, F., De Vries, H., Dumoulin, V., and Courville, A. Film: Visual reasoning with a general conditioning layer. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- Roweis, S. T. One microphone source separation. In *Advances in neural information processing systems*, pp. 793–799, 2001.
- Sablayrolles, A., Douze, M., Schmid, C., and Jégou, H. Spreading vectors for similarity search. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. URL <https://openreview.net/forum?id=SkGuG2R5tm>.
- Uhlich, S., Porcu, M., Giron, F., Enenkl, M., Kemp, T., Takahashi, N., and Mitsufuji, Y. 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)*, pp. 261–265. IEEE, 2017.
- van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. W., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. In *SSW*, pp. 125. ISCA, 2016.
- Vincent, E., Gribonval, R., and Févotte, C. Performance measurement in blind audio source separation. *IEEE Trans. Audio, Speech & Language Processing*, 14(4):1462–1469, 2006.
- Vincent, E., Virtanen, T., and Gannot, S. *Audio Source Separation and Speech Enhancement*. John Wiley Sons, Ltd, 2018. ISBN 9781119279860.
- Wan, L., Wang, Q., Papir, A., and Lopez-Moreno, I. Generalized end-to-end loss for speaker verification. In *ICASSP*, pp. 4879–4883. IEEE, 2018.
- Wang, D. and Chen, J. Supervised speech separation based on deep learning: An overview. *IEEE/ACM Trans. Audio, Speech & Language Processing*, 26(10):1702–1726, 2018.
- Wang, Q., Muckenhirn, H., Wilson, K. W., Sridhar, P., Wu, Z., Hershey, J. R., Saurous, R. A., Weiss, R. J., Jia, Y., and Lopez-Moreno, I. Voicefilter: Targeted voice separation by speaker-conditioned spectrogram masking. *CoRR*, abs/1810.04826, 2018a.
- Wang, Z., Roux, J. L., and Hershey, J. R. Alternative objective functions for deep clustering. In *ICASSP*, pp. 686–690. IEEE, 2018b.
- Wang, Z., Roux, J. L., Wang, D., and Hershey, J. R. End-to-end speech separation with unfolded iterative phase reconstruction. In *INTERSPEECH*, pp. 2708–2712. ISCA, 2018c.
- Weng, C., Yu, D., Seltzer, M. L., and Droppo, J. Deep neural networks for single-channel multi-talker speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23:1670–1679, 2015.
- Wichern, G., Antognini, J., Flynn, M., Zhu, L. R., McQuinn, E., Crow, D., Manilow, E., and Le Roux, J. Wham!+: Extending speech separation to noisy environments. In *Proc. Interspeech*, September 2019.
- Williamson, D. *Discrete-time Signal Processing: An Algebraic Approach*. Advanced Textbooks in Control and Signal Processing. Springer, 2012. ISBN 9781447105411.
- Xu, C., Rao, W., Xiao, X., Chng, E. S., and Li, H. Single channel speech separation with constrained utterance

- level permutation invariant training using grid LSTM. In *ICASSP*, pp. 6–10. IEEE, 2018.
- Yilmaz, O. and Rickard, S. Blind separation of speech mixtures via time-frequency masking. *IEEE Transactions on signal processing*, 52(7):1830–1847, 2004.
- Yu, D., Kolbæk, M., Tan, Z.-H., and Jensen, J. 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)*, pp. 241–245. IEEE, 2017.
- Yu, F. and Koltun, V. 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.
- Zeghidour, N., Synnaeve, G., Usunier, N., and Dupoux, E. Joint learning of speaker and phonetic similarities with siamese networks. In *INTERSPEECH*, pp. 1295–1299. ISCA, 2016.
- Zhang, L., Shi, Z., Han, J., Shi, A., and Ma, D. Furcanext: End-to-end monaural speech separation with dynamic gated dilated temporal convolutional networks. In *International Conference on Multimedia Modeling*, pp. 653–665. Springer, 2020.