Graph-PIT: Generalized permutation invariant training for continuous separation of arbitrary numbers of speakers

Thilo von Neumann[1], *Keisuke Kinoshita*[2], *Christoph Boeddeker*[1], *Marc Delcroix*[2], *Reinhold Haeb-Umbach*[1]

[1] Paderborn University, Germany
[2] NTT Corporation, Japan

{vonneumann,boeddeker,haeb}@nt.upb.de, {keisuke.kinoshita,marc.delcroix}@ieee.org

Abstract

Automatic transcription of meetings requires handling of overlapped speech, which calls for continuous speech separation (CSS) systems. The uPIT criterion was proposed for utterance-level separation with neural networks and introduces the constraint that the total number of speakers must not exceed the number of output channels. When processing meeting-like data in a segment-wise manner, i.e., by separating overlapping segments independently and stitching adjacent segments to continuous output streams, this constraint has to be fulfilled for any segment. In this contribution, we show that this constraint can be significantly relaxed. We propose a novel graph-based PIT criterion, which casts the assignment of utterances to output channels in a graph coloring problem. It only requires that the number of concurrently active speakers must not exceed the number of output channels. As a consequence, the system can process an arbitrary number of speakers and arbitrarily long segments and thus can handle more diverse scenarios. Further, the stitching algorithm for obtaining a consistent output order in neighboring segments is of less importance and can even be eliminated completely, not the least reducing the computational effort. Experiments on meeting-style WSJ data show improvements in recognition performance over using the uPIT criterion.
Index Terms: Continuous speech separation, automatic speech recognition, overlapped speech, permutation invariant training

1. Introduction

The automatic transcription of meetings has become a focus of research in recent years [1–4]. Conventional speech analysis systems, such as speech recognition, are constructed for a single active speaker at a time [5,6]. Since meeting recordings naturally contain overlapped speech, these systems cannot be applied directly, but require speech separation as pre-processing.

Many effective speech separation techniques have been proposed in the recent years based on neural networks, such as Deep Clustering [7] and models based on Permutation Invariant Training (PIT) [8–12]. Current state-of-the-art systems use the Utterance-level PIT (uPIT) [9] training scheme [10–12]. uPIT training works by assigning each speaker to an output channel of a speech separation network such that the training loss is minimized. This introduces the constraint that the number of speakers K must not exceed the number of output channels N in the speech segment to be processed.

As meetings can be of arbitrary length and can contain an arbitrary number of speakers, Continuous Speech Separation (CSS) [2], i.e., handling of arbitrarily long audio streams, is required. CSS can be realized by segmenting the input and processing the segments independently [2, 13]. Adjacent segments are aligned using a similarity measure in a so-called stitching process. It was shown that the number of output channels of the source separator can be fixed to, say, $N = 2$, although the total number of speakers K in a meeting may be much larger. That is because, if we select sufficiently small segments, we can assume that normally the number of speakers that appear in such a short segment becomes equal to or smaller than N, regardless of K. Viewed differently, this means that the constraint of uPIT effectively limits the segment size. Even for the example of a relatively short segment size of 2.4 s, more than 22 % of the segments contain more than two speakers in the CHiME-5 [14] evaluation dataset. In addition, the constraint of uPIT cannot be fulfilled when applied to or trained on full meetings.

In this contribution, we propose a generalization of uPIT, which relaxes of the above constraint ($K \leq N$). The generalization is achieved by incorporating the idea that different speakers can be put on the same output channel as long as they never overlap. We reformulate the problem of assigning utterances to output channels as a graph coloring problem, hence the name Graph-based Permutation Invariant Training (Graph-PIT). With Graph-PIT, we only need to ask for the number of *concurrently* active speakers, i.e., speakers speaking at the same time, to not exceed the number of output channels. This constraint is far more realistic and easier to satisfy compared to the original constraint imposed by uPIT. Looking again at the CHiME-5 evaluation dataset, this constraint is only violated 9 % of the time compared to 22 % for the uPIT constraint in case of $N = 2$. In a stitching-based CSS scenario, the proposed Graph-PIT criterion allows for arbitrarily long segments and arbitrarily many speakers in a segment as long as no more than N speakers speak at a time. Moreover, in a general CSS scenario without segmentation and stitching, Graph-PIT theoretically allows modeling the entire meeting with a separator that can utilize any contextual information.

In our experiments, we show the effectiveness of the proposed Graph-PIT loss on simulated meetings based on WSJ data. We can increase the segment size for stitching significantly and show that stitching is not even necessary for two-minute long meeting-like data.

2. Continuous Speech Separation

CSS [2] describes the task of separating an input audio signal $\mathbf{y}$ into one or multiple overlap-free signals $\hat{\mathbf{s}}_n, n = 1, ..., N$. We model a meeting $\mathbf{y}$ as the sum of U utterance signals produced by K different speakers, where utterances of different speakers may overlap:

$$\mathbf{y} = \sum_{u=1}^{U} \mathbf{s}_u. \tag{1}$$

The signal $\mathbf{s}_u$ is the u-th utterance, shifted and zero-padded to the length of the full meeting.

Attempts to solve the CSS problem led to multiple different approaches, e.g., yielding one signal per utterance ($N > K$) [15, 16] or producing one continuous stream per speaker [1, 17] ($K = N$). Here, we concentrate on the ideas proposed in [13].

The approach from [13] is based on the idea that the number of concurrently speaking speakers is usually much smaller than total number of speakers in a meeting, i.e., less output channels than speakers are required ($N \leq K$). It thus segments the meeting signal $\mathbf{y}$ into overlapping segments, separates the speech in each segment independently into N signals, and uses a stitching mechanism (Section 3.1) to align the output streams across segments. The neural-network-based separator is trained with uPIT (Section 2.1), which imposes the constraint that the number of speakers in a segment must not exceed the number of output channels, N.

We propose to replace uPIT with Graph-PIT (Section 2.2) to relax this constraint, only requiring that the number of simultaneously speaking speakers is smaller than the number of output channels. In the remainder of this section, we look at a segment of a meeting, i.e., K represents the number of speakers in the segment and $\mathbf{s}_u$ is scoped to this segment, to simplify explanations. We assume a time-domain source separator with N output channels.

2.1. Utterance-level PIT

The traditional uPIT [9] assigns each speaker to an output channel, i.e., $N = K$.[1] During training, the permutation problem between targets and estimated audio streams is solved by finding the bijective mapping $\pi^{(\text{uPIT})} : \{1, ..., K\} \to \{1, ..., N\}$ between speakers and output channels that minimizes the loss:

$$\mathcal{L}^{(\text{uPIT})} = \min_{\pi^{(\text{uPIT})} \in \mathcal{P}_N} \sum_{k=1}^{K} \mathcal{L}\big(\mathbf{s}_k^{(\text{spk})}, \hat{\mathbf{s}}_{\pi^{(\text{uPIT})}(k)}\big). \tag{2}$$

The loss function $\mathcal{L}$ is a signal-level loss function, $\mathcal{P}_N$ is the set of all permutations of length N and $\mathbf{s}_k^{(\text{spk})}$ is the sum of all utterances of speaker k.

2.2. Graph-based meeting-level PIT

If we relax the constraint $K = N$ of uPIT to $N \leq K$, i.e., an output channel is no longer bound to a speaker, there is no bijective mapping between output channels and speakers. We can, however, find an, in general, non-bijective mapping $\pi : \{1, ..., U\} \to \{1, ..., N\}$ of target utterances to output channels so that overlapped utterances are separated.

Finding such a mapping is equivalent to a graph coloring problem [18]; if each utterance is modelled as a vertex and edges are drawn between utterances that overlap, the set of all proper N-vertex-colorings of this graph is equal to the set of mappings from utterances to output channels. A N-vertex-coloring assigns each vertex a color from a set of N colors so that connected vertices should be colored differently.

This graph $G = (V, E)$ is undirected and defined with

$$\begin{aligned} V &= \{1, \dots, U\}, \\ E &= \{\{u, v\} : \forall u, v \in V, u \neq v \text{ if } \mathbf{s}_u \text{ and } \mathbf{s}_v \text{ overlap}\}, \end{aligned} \tag{3}$$

where V are the vertices / utterances and E the edges between overlapping utterances. An example of such a graph with two connected components is shown in the top part of Fig. 1. A proper N-vertex-coloring of G is defined as a mapping

$$\begin{aligned} &\pi^{(\text{Graph-PIT})} : V \to \{1, ..., N\}, \text{ such that} \\ &\pi^{(\text{Graph-PIT})}(u) \neq \pi^{(\text{Graph-PIT})}(v) \quad \forall \{u, v\} \in E. \end{aligned} \tag{4}$$

Note that $\pi^{(\text{Graph-PIT})}$ does not have to be surjective, i.e., the mapping is not required to use all output channels. Let $\mathcal{C}_{G,N}$ be the set of all proper N-vertex-colorings of G. Then, we formulate the Graph-PIT[2] objective as

$$\mathcal{L}^{(\text{Graph-PIT})} = \min_{\pi^{(\text{Graph-PIT})} \in \mathcal{C}_{G,N}} \sum_{n=1}^{N} \mathcal{L}\left(\tilde{\mathbf{s}}_{\pi^{(\text{Graph-PIT})}, n}, \hat{\mathbf{s}}_n\right), \tag{5}$$

where we construct an intermediate target signal

$$\tilde{\mathbf{s}}_{\pi, n} = \sum_{u=1}^{U} \begin{cases} \mathbf{s}_u, & \text{if } \pi(u) = n, \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{6}$$

The set $\mathcal{C}_{G,N}$ is computed by enumerating all graph colorings.

An example graph G together with a possible mapping of target utterances to output channels, i.e., a coloring, is drawn for a fictive speaker activity pattern in Fig. 1. Graph-PIT is equivalent to uPIT for a connected graph with two speakers only; an example of this is marked with (a). If the graph consists of more than one connected component (assuming $K \leq N$) uPIT places all utterances of a single speaker on the same output channel, which enforces the model to use global information. Graph-PIT gives more freedom to the placement as individual connected components are treated separately. If there are more than N speakers in a connected component (as in the second connected component of G in Fig. 1), Graph-PIT can provide a solution for the assignment problem while uPIT cannot.

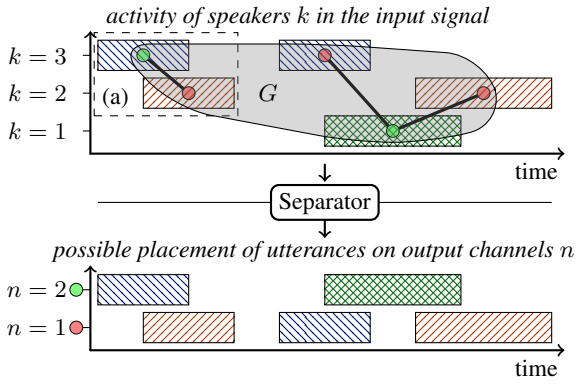

Figure 1: *Example of processing a three-speaker scenario using Graph-PIT with a two-output separator. Each box represents one utterance.* Top*: Utterances in the meeting and the colored overlap graph G. Graph-PIT is equivalent to uPIT for an activity pattern as marked with (a).* Bottom*: A possible mapping of utterances to output channels.*

2.3. Complexity

The vertex coloring problem is in general NP-hard [18]. While uPIT computes $N!$ permutations, the number of valid colorings for a graph is bounded by N^U (in the extreme case of no edges),

[1] This can be relaxed to $K < N$ by adding silent target signals $\mathbf{s}_i = 0$ for $K < i \leq N$.

[2] Code is available at `https://github.com/fgnt/graph_pit`.

which can easily exceed $N!$ if $U >> N$. However, N and U are typically small in a segment used for training and it is unlikely that the graph has no edges so that the computational overhead is negligible in comparison to the neural network.

2.4. Thresholded SDR for varying numbers of speakers

For this paper, we use a time-domain separation model as a basis of investigation. However, the Signal-to-Distortion Ratio (SDR)-based losses, which have shown to be effective for training such models [11, 12], are problematic when used in training with meeting-like data containing significant amounts of silence. Specifically, these losses have the problem that their value is unbounded so that easy examples can dominate the training and that they are undefined for silent target signals, i.e., $\mathbf{s} = \mathbf{0}$. Silent targets are required for training an N-output separator with less than N target utterances, e.g., when the mapping $\pi^{(\text{Graph-PIT})}$ is not surjective in Eq. (5). The first problem can be solved by the Thresholded SDR (tSDR) loss [19][3] which limits the value of the loss function at a soft maximum. The second problem can be solved by adding a small constant ε, ending up with the ε-tSDR loss

$$\begin{aligned}\mathcal{L}^{(\varepsilon\text{-tSDR})}(\mathbf{s}, \hat{\mathbf{s}}) &= -10 \log_{10} \frac{|\mathbf{s}|^2 + \varepsilon}{|\mathbf{s} - \hat{\mathbf{s}}|^2 + \tau(|\mathbf{s}|^2 + \varepsilon)} \\ &\geq -\text{SDR}_{\text{max}},\end{aligned} \tag{7}$$

where $\tau = 10^{-\text{SDR}_{\text{max}}/10}$ introduces soft threshold at SDR_{max}. The constant ε makes sure that the loss is defined even if $\mathbf{s} = \mathbf{0}$.

3. Experiments

In this section, we investigate separation networks trained with uPIT and Graph-PIT as meeting-level separators and segment-level separators in a stitching-based system. We expect that uPIT works only in the stitching-based case, while Graph-PIT works for batch-processing as well as segment-level separation.

3.1. Stitching for CSS

For the stitching-based system, we use a processing scheme similar to [2, 20]. The incoming audio is segmented into overlapping segments. Each segment is processed by the separation network independently. A segment consists of three subsegments representing a history, current and future context with lengths of T_h, T_c and T_f seconds, respectively. Segments are shifted by T_c. The current context is used for reconstruction while history and future contexts improve the separation quality at segment edges and are used to align output streams based on squared differences between overlapping sub-segments.

3.2. Data

For our experiments, we generate artificial meetings based on the WSJ [21] corpus. We randomly sample five to eight speakers for each meeting so that the distribution of speakers is uniform across all meetings. We also sample an overlap ratio between 0.2 and 0.4 for each meeting. Then, we uniformly select utterances of the sampled speakers and sample a start times so that the overlap ratio is roughly fulfilled and speakers are active for roughly the same amount of time. Short silence is sampled between two utterances with a probability of 10 %. All utterances of a speaker are scaled with the same logarithmic weight uniformly drawn from 0 dB to 5 dB. The meetings are corrupted by 20 dB to 30 dB of simulated white microphone noise.

The train and validation sets are based on *train_si284* and *cv_dev93* , respectively. The evaluation set is based on *test_eval92* with a total length of about 1 h. The meetings are about 120 s long which matches the length of segments used for meeting-wise evaluation in libri-CSS [2]. We intentionally stick to clean reverberation-free meetings based on WSJ instead of the libri-CSS database [2] for a proof-of-concept of the Graph-PIT objective. Libri-CSS is based on Librispeech [22] which contains utterances with large portions of silence. This makes it unclear how to define utterance boundaries. We use a sample rate of 8 kHz to speed up the experiments.

3.3. Metrics

For evaluation, we use metrics computed with an oracle diarization system. We perform separation in a continuous manner, i.e., we feed whole meetings into the separator, but compute the metrics utterance-wise by using oracle utterance boundaries. This scheme gives an upper bound on the performance. We compute the Word Error Rate (WER) and SDR [23] improvement (SDRi) compared to the unprocessed meeting.

We use an End-to-End ASR model from ESPnet [6] trained on clean WSJ data re-sampled to 8 kHz to compute the WER. The model achieves a WER of 5.6 % on the clean eval92 set of WSJ. We use the mir_eval toolbox [24] to obtain the SDRi.

3.4. Model architecture and training scheme

As the separation model, we use a Dual-Path Recurrent Neural Network (DPRNN)-based Time-domain Audio Separation Network (TasNet) [12] with two output channels, so we fix $N = 2$. To keep the computational cost low, we work with a smaller model with three blocks instead of six. The remaining configuration is the same as [12]: We set the number of filters in the encoder and decoder to 64, the number of hidden units to 128 in each direction, and the chunk length to 100. We use the ε-tSDR loss for all models with $\text{SNR}_{\text{max}} = 20\,\text{dB}$ and $\varepsilon = 10^{-6}$. Our architecture achieves 15 dB SDR gain on WSJ0-2mix [7].

3.4.1. Baseline uPIT model

We train the baseline system similar to [13] but with DPRNN-TasNet with uPIT on segments of meeting-like data, discarding any segments containing more than two speakers to match the evaluation data as closely as possible. We adjust the batch size so that all models see the same amount of 32 s of data per batch when training with different segment lengths T_tr.

3.4.2. Graph-PIT model

We train our proposed model with the following schedule: We train the model on segments of meeting-like data with uPIT like the baseline model which eases convergence at the beginning of training. Then, we re-train the pre-trained model with the Graph-PIT loss on the full training data including segments with more than two speakers. We reduce the amount of single-speaker examples during re-training to obtain a significant amount of examples with more than two speakers.

3.5. Meeting-wise evaluation

Table 1 shows the performance of different separation models on meetings with a length of about 120 s. We show results obtained with meeting-level batch processing as well as with stitching-based CSS. For stitching-based CSS, we only show the results for the best stitcher configuration for each model,

[3] tSDR is called "thresholded signal-to-noise ratio" in [19], but it measures distortions rather than noise in this case.

Table 1: *Performance on meetings of about* 120 s *length with five to eight speakers. Only the best stitcher configurations are shown. T_{tr} is the length of training segments in seconds. Experiments without stitching are marked with –. Best results are* ***bold*** *and the best results without stitching are* underlined.

Model	T_{tr}	Stitching	WER for num. spk			total	
		T_{h}+T_{c}+T_{f}	1	2	3	SDRi	WER
No sep.			10.9	45.9	67.4	0.0	49.1
uPIT	2	1+0.4+1	7.5	14.1	16.4	11.8	14.1
		–	10.1	28.3	38.8	4.6	30.0
	4	1+2+1	8.1	12.6	16.2	12.1	13.3
		–	9.7	27.7	41.6	3.9	30.3
	8	1+2+1	8.8	13.7	18.1	11.5	14.6
		–	8.0	17.7	26.9	8.9	19.7
	16	1+6+1	8.7	16.4	21.5	10.1	17.2
		–	8.0	17.3	23.6	9.3	18.4
Graph-PIT	4	1+1+1	7.5	**12.0**	**15.0**	**12.6**	**12.5**
		–	7.2	20.7	27.8	8.1	21.5
	8	1+6+1	7.7	12.8	16.1	12.3	13.2
		–	8.2	14.8	19.6	10.9	15.6
	16	1+6+1	9.0	13.2	16.4	11.9	13.8
		–	7.1	13.2	16.7	11.9	13.7
	32	1+6+1	7.8	12.5	16.9	12.0	13.5
		–	**7.0**	12.2	16.2	12.1	13.0

determined by keeping T_{h} and T_{f} constant at 0.4 s and sweeping T_{c} from 1 s to 14 s on the development set. We observed that the stitching process works reasonably well for $T_{\text{h}} = T_{\text{f}} = 1$ s. A comparison of different stitcher configurations is given in Section 3.6. We only report results for Graph-PIT where we expect differences to uPIT; for small segment sizes that are usually used for training of uPIT, the probability of seeing more than two speakers in a training and test segment is small.

Table 1 shows the WER grouped by number of overlapping speakers per utterance, and the WER and SDRi for the whole meetings. When looking at the overall performance, we see a slight improvement for training closer to the evaluation condition with Graph-PIT over uPIT when they are used in stitching based CSS framework. Models trained with Graph-PIT generalize to processing the whole meeting of 120 s length at once without using a stitcher (13.0 % WER) when trained with long enough segments while uPIT does not handle this case well (18.4 % WER). That is because uPIT is, by its constraint, not built to handle more than two speakers as it is the case for full meetings, while Graph-PIT trains for this scenario. Longer training segments match the evaluation scenario better.

When looking at the performance with respect to the number of speakers per utterance, we observe that the largest improvement of Graph-PIT over uPIT comes from utterances that overlap with two other speakers (i.e., num. spk. = 3). This is the scenario that Graph-PIT is explicitly trained for but uPIT is not. The improvement for non-overlapped utterances (i.e., num. spk. = 1) over no separation comes from suppressing the microphone noise. Shorter training segments are better if the model is applied with stitching, while longer training segments are better when evaluated without stitching, for both objectives.

3.6. The effect of stitching

Fig. 2 shows the effect of stitching on the performance in WER. The bottom plot shows the distribution of the number of speakers in a segment. The red line represents the amount of segments that fulfill the number-of-speaker constraint ($K \leq N$) of uPIT. As expected, the uPIT model gets degraded with larger segments where its constraint is violated. The performance of the Graph-PIT model, on the other hand, improves with increasing segment sizes and even generalizes to processing the whole meeting at once. Graph-PIT shows a better WER than uPIT for all scenarios.

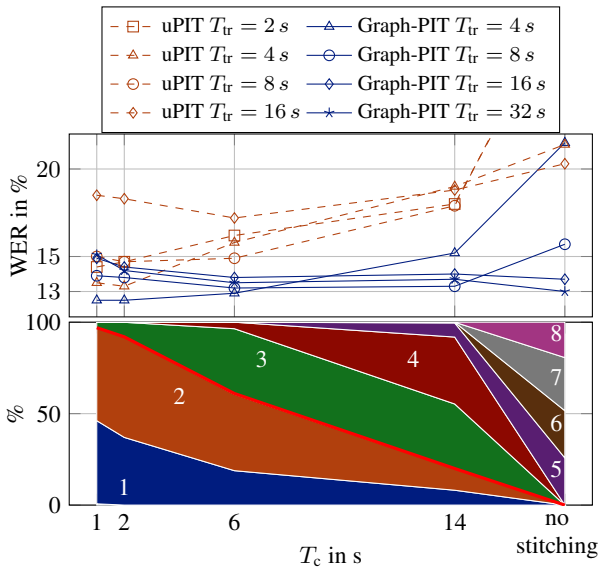

Figure 2: Top*: WER plotted over the segment size for stitching.* Bottom*: Distribution of the number of speakers in a segment. The red line represents the amount of segments that fulfill the constraint of uPIT.*

One advantage of larger segments for stitching is the reduced computational overhead that scales linearly with the segment overlap ratio. We can reduce the segment overlap ratio significantly from 200 % to 14 % by increasing the segment size from 3 s to 16 s and can eliminate the stitching process completely in our experiments on meetings with a length of 120 s.

It is interesting to see that the uPIT model, even though trained only on data with up to two speakers, does not break completely when it sees data violating its constraint. This can be seen in Fig. 2 and Table 1, i.e., 49.1 % WER with no processing and 18.4 % with batch processing. However, employing Graph-PIT during training drastically improves the performance in these cases.

4. Conclusions

In this paper, we proposed a generalization of uPIT for CSS-style processing of long recordings, called Graph-PIT. Graph-PIT relaxes the constraint of having less speakers than output channels in one segment to having less concurrent speakers (i.e., speakers active in one sample) than output channels. It thus enables processing of more diverse meetings and the use of much larger segments while reducing the computational overhead introduced by stitching. We showed that the Graph-PIT objective can be used to construct separation networks that do not require stitching at all. In future work, we plan to test Graph-PIT on more realistic data like libri-CSS [2] or real recordings.

5. Acknowledgements

Computational resources were provided by the Paderborn Center for Parallel Computing.

6. References

[1] T. v. 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 *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 91–95, iSSN: 2379-190X.

[2] Z. Chen, T. Yoshioka, L. Lu, T. Zhou, Z. Meng, Y. Luo, J. Wu, X. Xiao, and J. Li, "Continuous Speech Separation: Dataset and Analysis," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 7284–7288, iSSN: 2379-190X.

[3] S. Araki, N. Ono, K. Kinoshita, and M. Delcroix, "Comparison of Reference Microphone Selection Algorithms for Distributed Microphone Array Based Speech Enhancement in Meeting Recognition Scenarios," in *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, Sep. 2018, pp. 316–320.

[4] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner, "The AMI Meeting Corpus: A Pre-announcement," in *Machine Learning for Multimodal Interaction*, ser. Lecture Notes in Computer Science, S. Renals and S. Bengio, Eds. Berlin, Heidelberg: Springer, 2006, pp. 28–39.

[5] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," 2011. [Online]. Available: https://infoscience.epfl.ch/record/192584

[6] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-End Speech Processing Toolkit," in *Interspeech 2018*. ISCA, Sep. 2018, pp. 2207–2211.

[7] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 31–35.

[8] D. Yu, M. Kolbæk, Z. 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)*, Mar. 2017, pp. 241–245, iSSN: 2379-190X.

[9] M. Kolbæk, D. Yu, Z.-H. Tan, and J. Jensen, "Multitalker Speech Separation With Utterance-Level Permutation Invariant Training of Deep Recurrent Neural Networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1901–1913, Oct. 2017, conference Name: IEEE/ACM Transactions on Audio, Speech, and Language Processing.

[10] Y. Luo and N. Mesgarani, "TaSNet: Time-Domain Audio Separation Network for Real-Time, Single-Channel Speech Separation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 696–700, iSSN: 2379-190X.

[11] ——, "Conv-TasNet: Surpassing Ideal Time–Frequency Magnitude Masking for Speech Separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1256–1266, Aug. 2019, conference Name: IEEE/ACM Transactions on Audio, Speech, and Language Processing.

[12] 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)*, May 2020, pp. 46–50, iSSN: 2379-190X.

[13] T. Yoshioka, H. Erdogan, Z. Chen, X. Xiao, and F. Alleva, "Recognizing Overlapped Speech in Meetings: A Multichannel Separation Approach Using Neural Networks," in *Interspeech 2018*. ISCA, Sep. 2018, pp. 3038–3042. [Online]. Available: http://www.isca-speech.org/archive/Interspeech_2018/abstracts/2284.html

[14] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The Fifth 'CHiME' Speech Separation and Recognition Challenge: Dataset, Task and Baselines," in *Interspeech 2018*. ISCA, Sep. 2018, pp. 1561–1565.

[15] N. Kanda, S. Horiguchi, Y. Fujita, Y. Xue, K. Nagamatsu, and S. Watanabe, "Simultaneous Speech Recognition and Speaker Diarization for Monaural Dialogue Recordings with Target-Speaker Acoustic Models," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Dec. 2019, pp. 31–38.

[16] S. Watanabe, M. Mandel, J. Barker, E. Vincent, A. Arora, X. Chang, S. Khudanpur, V. Manohar, D. Povey, and D. Raj, "CHiME-6 Challenge: Tackling multispeaker speech recognition for unsegmented recordings," in *6th International Workshop on Speech Processing in Everyday Environments (CHiME 2020)*, 2020.

[17] T. Hori, S. Araki, T. Yoshioka, M. Fujimoto, S. Watanabe, T. Oba, A. Ogawa, K. Otsuka, D. Mikami, K. Kinoshita, T. Nakatani, A. Nakamura, and J. Yamato, "Low-Latency Real-Time Meeting Recognition and Understanding Using Distant Microphones and Omni-Directional Camera," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 499–513, Feb. 2012, conference Name: IEEE Transactions on Audio, Speech, and Language Processing.

[18] J. E. Hopcroft, R. Motwani, and J. D. Ullman, "Automata theory, languages, and computation," *International Edition*, vol. 24, no. 2, 2006.

[19] S. Wisdom, E. Tzinis, H. Erdogan, R. J. Weiss, K. Wilson, and J. R. Hershey, "Unsupervised Speech Separation Using Mixtures of Mixtures," in *ICML 2020 Workshop on Self-Supervision for Audio and Speech*, 2020.

[20] T. Yoshioka, H. Erdogan, Z. Chen, and F. Alleva, "Multi-Microphone Neural Speech Separation for Far-Field Multi-Talker Speech Recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 5739–5743, iSSN: 2379-190X.

[21] J. Garofalo, D. Graff, D. Paul, and D. Pallett, "Csr-i (wsj0) complete," *Linguistic Data Consortium, Philadelphia*, 2007.

[22] 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)*, Apr. 2015, pp. 5206–5210, iSSN: 2379-190X.

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

[24] C. Raffel, B. Mcfee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. W. Ellis, C. C. Raffel, B. Mcfee, and E. J. Humphrey, "mir_eval: A Transparent Implementation of Common MIR Metrics," in *In Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR*, 2014.