Many-Speakers Single Channel Speech Separation
with Optimal Permutation Training

Shaked Dovrat[1]*, Eliya Nachmani[1,2]*, Lior Wolf[1]

[1]Blavatnik School of Computer Science, Tel Aviv University, Israel
[2]Facebook AI Research, Israel

shaked.dovrat@gmail.com, enk100@gmail.com, wolf@cs.tau.ac.il

Abstract

Single channel speech separation has experienced great progress in the last few years. However, training neural speech separation for a large number of speakers (e.g., more than 10 speakers) is out of reach for the current methods, which rely on the Permutation Invariant Training (PIT). In this work, we present a permutation invariant training that employs the Hungarian algorithm in order to train with an $O(C^3)$ time complexity, where C is the number of speakers, in comparison to $O(C!)$ of PIT based methods. Furthermore, we present a modified architecture that can handle the increased number of speakers. Our approach separates up to 20 speakers and improves the previous results for large C by a wide margin.

Index Terms: speech separation, single channel, deep learning

1. Introduction

Single channel speech separation is a fundamental problem in the speech process, which has seen tremendous advances in the last years. The main neural architectures can be divided into two categories: (i) Spectral based [1, 2, 3] and (ii) time domain based [4, 5, 6]. Currently, the latter category leads with respect to the obtained accuracy.

Since the order of the speakers at the output of the neural network is arbitrary, a permutation invariant training is performed. Most of the neural architecture for speech separation use the permutation invariant training (PIT) loss [7] or its extension to the utterance level (uPIT) [8]. Both variants have a computational complexity of $O(C!)$, where C is the number of the speakers. As a result, it is not feasible to run PIT on more than ten speakers.

In this work, following [9] and [10], we propose a novel method to train a large number of speakers with a lower complexity of $O(C^3)$, by using the Hungarian algorithm. The Hungarian algorithm is able to find the optimal permutation in terms of the minimal sum of pairwise losses, thus matching between pairs of output- and target-signals. In order to enable the separation network to deal with a large number of speakers, we further introduce an architecture that combines two distinct approaches to separation networks, LSTM and dilated convolutional layers.

In our experiments, our method separates up to 20 speakers, which, as far as we can ascertain, is twice the number tackled by any existing method. Moreover, we show that our method improves the previous state of the art separation results for separating 5 and 10 speakers.

2. Related Work

Signal channel speech separation was explored using classical approaches [11, 12] and, more recently, using deep learning

---

*Equal contribution

methods. In [13] an LSTM neural network with a phase sensitive loss function was introduced. An improvement in SDR was demonstrated on the CHiME-2 [14] dataset. In [15] a neural separation network with a clustering-based embedding was introduced, presenting results for the separation of two speakers and introducing the WSJ-2mix dataset that was extensively used by followup work. This work was further extended in [16] by extracting an embedding of spectrogram segments and estimating a mask for the separation part. Results were provided for two and three speakers and an SDR improvement of 10.3 dB and 7.1 dB for WSJ-2mix and WSJ-3mix was obtained. In [17], the neural separator network was introduced. Attractor points in the embedding space were used to obtain the time-frequency bins for each speaker. The improvement on the WSJ-mix dataset was by 5.49%.

Luo et al. [4] introduced TasNet, which is a time domain encoder-decoder neural architecture for the single channel speech separation problem. They show a results of 11.1 SDR improvement for WSJ-2mix dataset over the state of the art. Wang et al. [3] proposed a neural architecture that separates the speakers in both the time and the frequency domains simultaneously. They presented an improvement of 13.2 SDR on the WSJ-2mix dataset. The work of [4] further improved the architecture and introduced ConvTasNet [18], which employed a dilated convolutional neural network, showing an SDR improvement 15.6 dB for the WSJ-2mix dataset.

Another improvement with LSTM network was introduced in [19], where the dual-path recurrent neural network (DPRNN) architecture was employed to model extremely long sequences. They showed SDR improvement of 18.08 dB on WSJ-2mix dataset. In [5] a separation network with $MulCat$ blocks was introduced. The proposed method also removed the masking sub-network, leading to an improvement of 20.12 dB to WSJ-2mix dataset. Furthermore, the WSJ-mix dataset was extended to include mixtures of 5 speakers, where the SDR improvement was 10.6 dB. [20] combined DPRNN and TasNet and for the WSJ-5mix dataset they showed an SDR improvement of 10.41dB, and 11.14dB SDR improvement for online remixing. Zeghidour et al. [6] introduced a neural separation network that infers a representation to each speaker, by performing clustering, and used it to separate the mixture. They show an SDR improvement of 22.2 dB for WSJ-2mix dataset.

Since our work builds upon the $MulCat$ network architecture [5], we will recap its major components. The network consists of an encoder, a separation module and a decoder. The encoder and decoder are simple 1D convolutions. The separation module starts with a chunking module which cuts the signal into chunks in time. Then, a series of doubled $MulCat$ blocks is applied. During training, a multi-scale loss is employed- after each doubled $MulCat$ block the activations are reconstructed by the decoder into audio signals and fed into the loss function.

The method also uses the Scale-Invariant Signal-to-Noise Ratio (SI-SNR) loss, which is a slight improvement to the traditional SDR loss.

Another line of work for speech separation uses beamformers and introduces an extension of the minimum variance distortionless response (MVDR) [21]. A neural beamformer was introduced in [22] and further improved in [23] for the speech separation problem. A follow up work introduced the linearly constrained minimum variance (LCMV) beamformer [24].

In [25] a SinkPIT loss is introduced. They proposed a variant of the PIT loss, which is based on Sinkhorn's matrix balancing algorithm. They reduce the complexity of the PIT loss from $O(C!)$ to $O(kC^2)$, where k is set to 200. It is important to note that the chosen permutation is only an approximation of the optimal permutation. In another work, a probabilistic-PIT loss which considers the output permutation as discrete latent random variable was introduced [26]. [9] and [10] noted that it is possible to use the Hungarian algorithm to find the best permutation for source separation.

2.1. Hungarian Algorithm

The *linear sum assignment* problem (also known as the *assignment problem*) is the task of assigning C agents to do C tasks, such that each agent is assigned to exactly one task, and the total cost for the agents performing the tasks is minimal. In other words, given a C-by-C matrix of costs, M, one for each agent-task pair, find a permutation π of the agents, such that the sum of costs of paired agents and tasks is minimal:

$$\pi = \underset{\pi \in \Pi_C}{\operatorname{argmin}} \sum_{i=1}^{n} M_{i,\pi(i)} \qquad (1)$$

A naive solution is to iterate over all $C!$ possible permutations. Fortunately, an optimal and polynomial-time algorithm that solves the assignment problem was proposed in 1955 by Harold Kuhn [27], reviewed in 1957 by James Munkres [28] and is mostly known by the name the *Hungarian Algorithm*. The initial time complexity of the algorithm was $O(C^4)$ and it was modified later on to a time complexity of $O(C^3)$ [29].

Simply put, the algorithm starts by trying to find an obvious permutation. If that fails, it goes on to make modifications to the input matrix, in order to find a valid and optimal permutation. The number of the modification iterations needed to find the solution is indicative of how well the permutation fits the data compared to alternative permutations. In other words, the more iterations needed until convergence, the less significant is the optimal permutation.

3. Method

This work extends the work in [5], through a number of new contributions. First, we introduce the Hungarian Loss which replaces the PIT loss and gives an optimal solution to the permutation issue with a much lower time complexity, $O(C^3)$, which allows to train separation networks for many speakers. Second, we introduce a new network architecture that uses stacked dilated convolutions before each pair of $MulCat$ blocks of [5]. The overall method, is depicted in Figure 1.

3.1. Hungarian Loss

A single-channel speech separation network takes an audio signal that contains a mixture of C speakers speaking and outputs C audio signals, each optimized to contain a separate speaker.

During training, the network outputs the separated audio

signals in an arbitrary order. Thus, in order to compute a meaningful loss, an alignment, i.e. a permutation, needs to be recovered between the outputs of the network and the separated target signals. One way to find the right permutation is to iterate over all possible $C!$ permutations and choose the one which gives the lowest mean loss value on the pairs (i.e. PIT). The computational cost of PIT is unnoticeable when C is small, in comparison to the other parts of the network. However, it makes training on a large number of speakers impossible. For instance, for 20 speakers PIT needs to check $20! \approx 2.4 \times 10^{18}$ different permutations).

To address this issue, we formulate the task of finding the permutation which minimizes the loss function as a *linear sum assignment* problem. Given the C output signals and C target signals, we calculate the pairwise loss value, $\hat{\ell}(s_i, \hat{s}_j)$, on every pair of output ($\hat{s}_j$) and target (s_i) signals, which gives an C-by-C matrix of losses, M. Next, we assign each output with a unique target and vice-versa. Such assignment is equivalent to choosing C elements of the matrix, such that each chosen element is in a different row and column from all others. An optimal assignment minimizes the sum of values of the chosen elements. The PIT loss can then be viewed as a brute-force solution to this problem, iterating over all possible solutions:

$$\ell(s, \hat{s}) = \min_{\pi \in \Pi_C} \frac{1}{C} \sum_{i=1}^{C} \hat{\ell}(s_i, \hat{s}_{\pi(i)}) \qquad (2)$$

By running the Hungarian Algorithm on M, we efficiently find the optimal permutation in polynomial time instead of the brute-force, factorial-time PIT. Note that the assignment algorithm does not need to be differential, since we find a permutation of the targets by which we calculate the loss, meaning that the process is separate from the backwards calculation of gradients.

3.2. Model

We shift our focus to solving the task of separating mixtures of many ($C \geq 10$) speakers. As C increases, the task of separating the mixtures becomes more challenging. Thus, we propose a new and suitable network architecture.

For this end, we modify the $MulCat$-based architecture [5] by adding stacked dilated convolutions before each pair of $MulCat$ blocks. In addition, we increased some of the network's hyper-parameters to achieve a larger capacity needed for the harder tasks. The dilated convolutions scheme is borrowed from [18]: We use 8 *1-D Conv block*s stacked on top of each other, with dilation factors 2^{i-1} for $i \in \{1, 2, ..., 8\}$. This corresponds to a single column in the separation module of [18].

The rest of the model is in accordance with [5], i.e. using the same encoder, chunking, $MulCat$ blocks, decoder and SI-SNR loss. We also adopt the multi-scale loss scheme, which applies the loss function after each double MulCat block, instead of just the last. We tuned some hyper-parameters of the architecture to accompany the harder tasks when C is large: N, number of features, was increased from 128 to 256. L, the encoder's kernel size was increased from 8 to 16. H, the number of hidden units in the LSTMs was increased from 128 to 256 and finally R, the number of double $MulCat$ blocks was increased from 6 to 7. A similar hyper-parameters adjustment was done in [25].

4. Experiments

Comparison to state of the art   We show results on datasets derived from WSJ corpus [30] and LibriSpeech [31]. For WSJ,
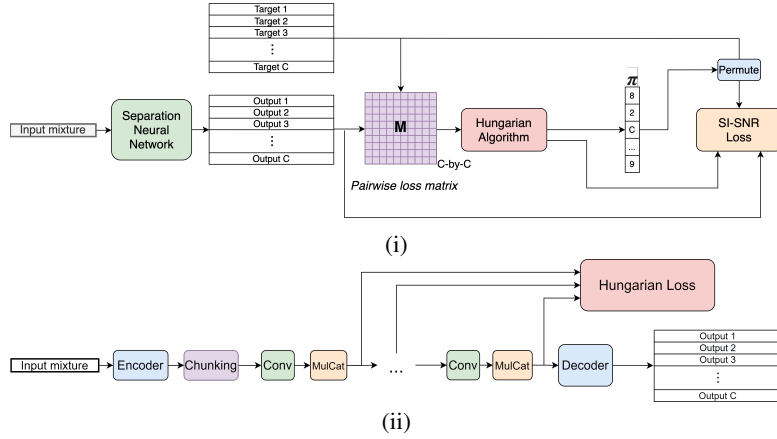
Figure 1: *(i) The proposed Hungarian Loss. M, a C-by-C matrix of SI-SNR losses between output and target pairs is computed. M is fed into the Hungarian algorithm, which efficiently finds the optimal permutation of target signals, π. (ii) The proposed separation network architecture. The novel components are the added Conv blocks and the Hungarian loss, which replaces the PIT loss.*
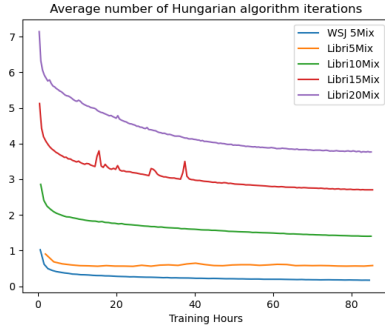

Figure 2: *The average number of iterations per example performed by the Hungarian algorithm. The average is decreasing as training progresses, which indicates that the separation is improving. This also means that the runtime of the Hungarian algorithm is getting even shorter.*

we use the 5-speaker mix, introduced in [5], which uses the same procedure as in [15], i.e. 30 hours of speech from the training set si_tr_s were used to create the training and validation sets. The five speakers were randomly chosen and combined with random SNR values between $0 - 5$ dB. The test set is created from si_et_s and si_dt_s with 16 speakers, that differ from the speakers of the training set. For LibriSpeech, we use the LibriMix [32] datasets with mixes of 5, 10, 15 and 20 speakers. LibriMix offers mixtures of 2 and 3 speakers from LibriSpeech, and we used the given scripts to create mixtures of 5, 10, 15 and 20 speakers. We used LibriMix's given parameters to get a sample rate of 8Khz, clean mixtures (no noise added) and each sample is cut in length, according to the minimal length sample in the mixture. We also use the augmentation process as in [25].

A separate model is trained for each dataset, with the corresponding number of output channels. Training was done using the Adam optimizer [33], with batch size 32 and a learning rate of $1e - 3$ which was multiplied by 0.95 every two epochs. During training, each sample is cut into 4-second segments. Table 1 compares the results of our model to other models, using the SI-SDRi metrics. As can be seen, our model outperforms the other methods by a large margin. For Libri-5mix and Libri-10mix, we improve the previous results by 1.89dB and 1.33dB respectively. Interestingly, the previous state of

the art results for $Libri5Mix$ is obtained with $MulCut$ [5], whereas for $Libri10Mix$ it is obtained by [25], which uses SinkPIT. This is due to the fact that the PIT loss in $MulCut$ for 10 speakers is prohibitive. For WSJ-5mix, the SDR of our method improves by more than 2dB over the previous method. We are the first to present results for the $Libri15Mix$ and $Libri20Mix$ datasets and running previous work on it is not practical. Sample results are shared online `https://shakeddovrat.github.io/hungarian/`.

Hungarian loss vs. alternative losses In order to show the benefits of using the Hungarian algorithm as opposed to PIT, we show the training duration of an epoch of each dataset, depicted in Table 2. As shown, on a 5 speaker mix both methods take about the same time. However, for $C = 10$, the Hungarian method is about 9 times faster on our model. For $C \geq 15$, the Hungarian method is still fast, while PIT is unusable as it failed to complete a single epoch after days of running. To our knowledge, the SinkPIT can run on 20 speakers but only find an approximation to the optimal permutation, whereas our method is both optimal and fast (the source code was not published).

In Figure 2, we present the average number of iterations per example that were performed by the Hungarian algorithm for the various datasets. We can observe two phenomena: (i) As the training proceeds and the neural network improves the separation performance, the average number of iterations is decreased. (ii) The average number of iterations is lower when the number of speakers in the mixture is lower. Both of these observations came from the fact that the Hungarian algorithm needs no iterations to converge when the outputs or the network are mixed or noisy.

In Figure 3, we plot the mel spectrogram for typical samples as in [25]. This is shown for a mixture of 10 speakers, from the $Libri10Mix$ dataset. As can be seen in speakers 4 and 8 our method provide a cleaner mel spectrogram, Furthermore, the SI-SDRi difference is 4.9 dB and 2.4 dB for speakers 4 and 8 respectively (there is a significant improvement in all speakers).

In Figure 4, we plot the pairwise SI-SDR negative matrix M sorted in descending order, in comparison to the results of the SinkPIT system [25]. Evidently, the entropy in our method is lower, especially in the last rows, i.e., it has much less confusion compared to the baseline method.

Ablation study We run ablation analysis in order to understand the contribution of each component of our method. The
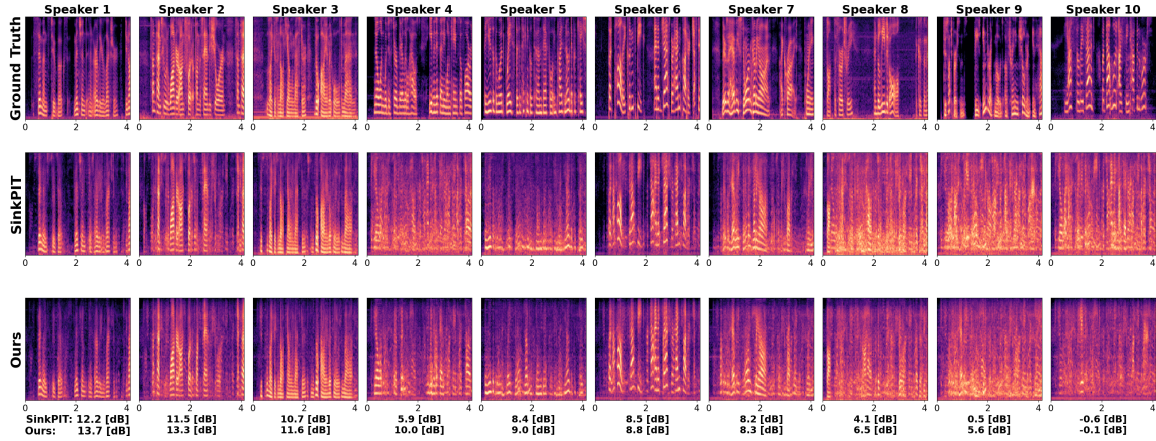
Figure 3: *Separation results for a mixture of* 10 *speakers. First row: mel spectrogram of ground truth signals. Second row: mel spectrogram of SinkPIT signals. Third row: mel spectrogram of the outputs of our method. The last row shows the SI-SDR improvement for the SinkPIT method and our method. The x-axis is sorted by the SI-SDRi in descending order.*
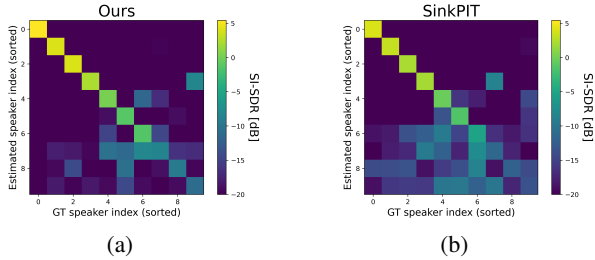

(a)   (b)

Figure 4: *The pairwise SI-SDR matrix M, sorted in descending order. For the same input sample as in Figure 3. (a) Our method. (b) SinkPIT [25].*

results are summarized in Table 3, where -C means without the 1D convolutions and -H means using the PIT loss instead of the Hungarian loss. As can be seen, adding the dilated convolution to the architecture improves the performance. Moreover, without the Hungarian algorithm, it is practically impossible to train on 15 or more speakers. For 10 speakers, using PIT drastically diminishes performance due to longer training time. On the other hand, for 5 speakers, the PIT and Hungarian have similar performance since they both found the optimal permutation in a similar runtime.

5. Conclusions

In this work, we provide a method for single channel sound separation for a large number of sources. Our method is the first work to show that one can separate a mixture of 20 speakers from a single channel recording. Our solution is based on the Hungarian algorithm, which efficiently finds the optimal permutation from the $C!$ possible permutations and on a new network architecture that adds stacked 1-D convolutions and added capacity to the state of the art architecture.

6. Acknowledgments

Table 1: *SDR improvement performance of various models versus number of speakers and datasets. 'X' results indicated simulation that failed to complete a single epoch, due to high complexity PIT loss. xMix is LibriMix with x speakers.*

Model	5Mix	10Mix	15Mix	20Mix	WSJ-5
ConvTasNet	-	-	-	-	6.8
DPRNN [19]	-	-	-	-	8.6
MulCat [5]	10.83	4.74	X	X	10.6
TasTas [20]	-	-	-	-	11.14
SinkPIT [25]	9.39	6.45	-	-	-
Ours	**12.72**	**7.78**	**5.66**	**4.26**	**13.22**

Table 2: *Training run times of PIT and Hungarian algorithm in minutes per epoch. 'X' indicates simulation that failed to complete a single epoch, due to long run times of the PIT loss.*

Dataset	#Spkrs	#Perms	PIT	Hungarian
WSJ-5mix	5	120	65	62
Libri-5Mix	5	120	139	140
Libri-10Mix	10	$\approx 3.6e6$	462	52
Libri-15Mix	15	$\approx 1.3e12$	X	36
Libri-20Mix	20	$\approx 2.4e18$	X	29

Table 3: *Ablation analysis - SDR Performance for LibriMix and WSJ-mix datasets. xMix is LibriMix with x speakers. "-C" means without the 1D convolutions and "-H" means using PIT instead of the Hungarian loss. 'X' indicates simulation that failed to complete a single epoch, due to long run times of the PIT loss.*

Model	5Mix	10Mix	15Mix	20Mix	WSJ-5
Ours-C-H	11.10	4.47	X	X	12.18
Ours-H	12.53	4.84	X	X	13.07
Ours-C	11.19	5.89	5.14	4.10	12.10
Ours	**12.72**	**7.78**	**5.66**	**4.26**	**13.22**

7. References

[1] Z.-Q. Wang, K. Tan, and D. Wang, "Deep learning based phase reconstruction for speaker separation: A trigonometric perspective," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 71–75.

[2] X. Li, X. Wu, and J. Chen, "A spectral-change-aware loss function for dnn-based speech separation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6870–6874.

[3] Z.-Q. Wang, J. L. Roux, D. Wang, and J. R. Hershey, "End-to-end speech separation with unfolded iterative phase reconstruction," *arXiv preprint arXiv:1804.10204*, 2018.

[4] 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)*. IEEE, 2018, pp. 696–700.

[5] E. Nachmani, Y. Adi, and L. Wolf, "Voice separation with an unknown number of multiple speakers," in *International Conference on Machine Learning*. PMLR, 2020, pp. 7164–7175.

[6] N. Zeghidour and D. Grangier, "Wavesplit: End-to-end speech separation by speaker clustering," *arXiv preprint arXiv:2002.08933*, 2020.

[7] 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.

[8] 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, 2017.

[9] N. Kanda, Y. Gaur, X. Wang, Z. Meng, and T. Yoshioka, "Serialized output training for end-to-end overlapped speech recognition," *arXiv preprint arXiv:2003.12687*, 2020.

[10] T. Ma, "Monaural source separation in the wild," Ph.D. dissertation, Carnegie Mellon University Pittsburgh, PA, 2020.

[11] R. Martin and I. Cohen, "Single-channel speech presence probability estimation and noise tracking," *Audio Source Separation and Speech Enhancement*, pp. 87–106, 2018.

[12] O. Ernst, S. E. Chazan, S. Gannot, and J. Goldberger, "Speech dereverberation using fully convolutional networks," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 390–394.

[13] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 708–712.

[14] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second 'chime'speech separation and recognition challenge: Datasets, tasks and baselines," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 126–130.

[15] 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.

[16] Y. Isik, J. L. Roux, Z. Chen, S. Watanabe, and J. R. Hershey, "Single-channel multi-speaker separation using deep clustering," *arXiv preprint arXiv:1607.02173*, 2016.

[17] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 246–250.

[18] Y. Luo and N. Mesgarani, "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, 2019.

[19] 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.

[20] A. Shi, J. Han, and Z. Shi, "Toward the pre-cocktail party problem with tastas +," *arXiv preprint arXiv:2009.03692*, 2020.

[21] S. Markovich, S. Gannot, and I. Cohen, "Multichannel eigenspace beamforming in a reverberant noisy environment with multiple interfering speech signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1071–1086, 2009.

[22] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, M. Bacchiani *et al.*, "Speaker location and microphone spacing invariant acoustic modeling from raw multichannel waveforms," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 30–36.

[23] Y. Luo, E. Ceolini, C. Han, S.-C. Liu, and N. Mesgarani, "Fasnet: Low-latency adaptive beamforming for multi-microphone audio processing," *arXiv preprint arXiv:1909.13387*, 2019.

[24] B. Laufer-Goldshtein, R. Talmon, and S. Gannot, "Global and local simplex representations for multichannel source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 914–928, 2020.

[25] H. Tachibana, "Towards listening to 10 people simultaneously: An efficient permutation invariant training of audio source separation using sinkhorn's algorithm," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021.

[26] M. Yousefi, S. Khorram, and J. H. Hansen, "Probabilistic permutation invariant training for speech separation," *arXiv preprint arXiv:1908.01768*, 2019.

[27] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[28] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the society for industrial and applied mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[29] N. Tomizawa, "On some techniques useful for solution of transportation network problems," *Networks*, vol. 1, no. 2, pp. 173–194, 1971.

[30] J. Garofolo, D. Graff, D. Paul, and D. Pallett, "Csr-i (wsj0) complete ldc93s6a," *Web Download. Philadelphia: Linguistic Data Consortium*, vol. 83, 1993.

[31] 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.

[32] 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.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.