



Speech Separation for an Unknown Number of Speakers Using Transformers With Encoder-Decoder Attractors

Srikanth Raj Chetupalli, Emanuël A. P. Habets

International Audio Laboratories Erlangen*, Am Wolfsmantel 33, 91058 Erlangen, Germany

srikanth.chetupalli@iis.fraunhofer.de, emanuel.habets@audiolabs-erlangen.de

Abstract

Speaker-independent speech separation for single-channel mixtures with an unknown number of multiple speakers in the waveform domain is considered in this paper. To deal with the unknown number of sources, we incorporate an encoder-decoder attractor (EDA) module into a speech separation network. The neural network architecture consists of a trainable encoder-decoder pair and a masking network. The mask network in the proposed approach is inspired by the transformer-based SepFormer separation system. It contains a dual-path block and a triple path block, each block modeling both short-time and long-time dependencies in the signal. The EDA module first summarises the dual-path block output using an LSTM encoder and generates one attractor vector per speaker in the mixture using an LSTM decoder. The attractors are combined with the dual-path block output to generate speaker channels, which are processed jointly by the triple-path block to predict the mask. Further, a linear-sigmoid layer, with attractors as the input, predicts a binary output to indicate a stopping criterion for attractor generation. The proposed approach is evaluated on the WSJ0-mix dataset with mixtures of up to five speakers. State-of-the-art results are obtained in the speech separation quality and speaker counting for all the mixtures.

Index Terms: source separation, speaker counting, attractors, transformers

1. Introduction

Single-channel speech separation, the task of estimating individual speech source signals from a single-channel mixture signal, is of interest for different speech technologies such as automatic speech recognition of real-world multi-speaker conversations, speech communication, speech archival, and indexing. The task is considered challenging because of the statistical similarities between the speech from different speakers and furthermore difficult when the number of speakers in the recording is not known a-priori.

Traditional approaches used non-negative matrix factorization [1, 2], computational auditory scene analysis [3, 4], and eigenvoice speaker modeling [5], etc. With the advent of deep learning, supervised approaches have gained significant interest. Early approaches involved learning to separate using time-frequency domain masking [6, 7]. A time-domain approach was proposed in [8, 9], in which a trainable encoder is used to convert the time-domain signal into a time-feature (TF) domain, the separating source masks are estimated in the TF domain, and a trainable decoder then reconstructs the separated signals. The approach is extended using recurrent networks, short-time and long-time processing for efficient long-sequence modeling in

*A joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer IIS, Germany.

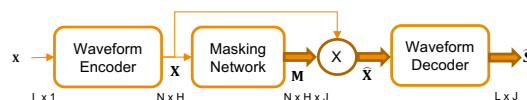


Figure 1: Block diagram of the overall separation scheme.

dual-path RNN (DPRNN) [10]. Several successful approaches employed the dual-path approach for source separation [11–14]. In SepFormer [14], the RNN layers in DPRNN were replaced with more efficient transformer layers and achieved state-of-the-art (SOTA) performance. The above approaches do not rely on any speaker/source representation. In contrast, source representations are obtained first, augmented with input, and then fed to a convolutional separation stack in [15]. In [9–15], the number of speakers in the mixture is assumed to be known and fixed, which may not hold in practice for real-world mixtures.

Speech separation for an unknown number of sources is also explored in the literature [16–18]. Using a one-and-rest permutation invariant training, a recursive source separation scheme has been proposed in [16]. In [17], a separate model was trained for every possible number of speakers. An activity detector on the model outputs is then employed to decide the number of speakers. Finally, the model output with the highest number of speakers is used as the output. Recently, a multi-decoder approach was proposed in [18] in which different decoders were trained corresponding to a different number of speakers with a shared encoder. The encoder output is also fed to a speaker-count head, which estimates the number of speakers. The decoder head corresponding to the estimated number of speakers is used during inference. The aforementioned architectures use multiple models [17], multiple decoders [18] or multiple forward passes through the network [16]. Instead, this paper proposes a single architecture for the source separation of an unknown number of sources.

The proposed architecture is inspired by the SepFormer [14] and uses LSTM encoder-decoder-based attractor calculation method proposed in [19] to deal with the unknown number of speakers. We show that SOTA performance is obtained on all the WSJ0-mix datasets with two-five speakers in the recording, using a network that is half the size of the current SOTA on WSJ0-mix for source separation.

2. Proposed Approach

Let \mathbf{x} denote a single-channel mixture of signals $\mathcal{S} = \{s_1, s_2, \dots, s_J\}$ from J number of speech sources. The goal in the present work is to obtain an estimate ($\hat{\mathcal{S}}$) of the set of source signals (\mathcal{S}) and the number of sources (J) given the mixture signal \mathbf{x} . To achieve this goal, we consider a supervised learning approach in which a neural network is trained to learn a mapping between the mixture signal and the unknown number of source signals.

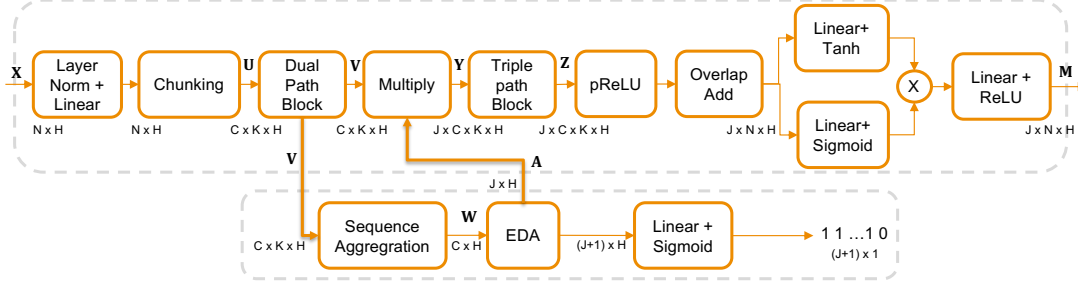


Figure 2: Block diagram of the masking network. The dimensions of the tensors at different stages of processing are indicated below the blocks.



Figure 3: Dual-path block.

2.1. Overview

We consider the encoder, masking network, and the decoder framework shown in Fig. 1. The waveform encoder converts the raw mixture speech of length L samples into a non-negative, sub-sampled, time-feature representation of dimension H with N time frames. The masking network predicts a non-negative mask for each source in the mixture, which is multiplied element-wise with the encoder output and fed to the waveform decoder to compute the individual source signals. Unlike many traditional approaches, which assume the number of sources J to be known in advance, the masking network in the present approach estimates the number of sources in the mixture. Individual components of the proposed architecture are described in the following sections.

2.2. Waveform Encoder-Decoder

The waveform encoder is composed using a 1-D convolution layer with rectified linear unit (ReLU) activation. The encoder takes L input samples ($\mathbf{x} \in \mathbb{R}^L$) and generates a time-feature (TF) representation ($\mathbf{X} \in \mathbb{R}_+^{N \times H}$). The convolution layer uses $H = 256$ filters of kernel size 16×1 and a stride of 8 samples.

The waveform decoder is a transposed-convolution layer, symmetric to the encoder, i.e., with the same number of filters, kernel size, and stride parameters as the encoder. Waveform decoder takes the masked TF representations ($\widehat{\mathbf{X}}_j \in \mathbb{R}_+^{N \times H}$) as input and computes the separated sources $\{\widehat{\mathbf{s}}_j \in \mathbb{R}^L, \forall j\}$.

2.3. Masking Network

The masking network, shown in Fig. 2, is inspired by the SepFormer [14] architecture but differs from SepFormer in the attractor calculation and the subsequent triple-path processing.

Input processing: The input TF representation \mathbf{X} is first passed through a layer-norm layer followed by a linear layer without bias, and segmented into overlapping (50% overlap) chunks of size $K = 250$. The 3D output of the chunking stage \mathbf{U} , of C chunks, is then input to the dual-path block.

Dual-path block: The dual-path block in the proposed work (Fig.3) is identical to the SepFormer block defined in [14]. It comprises of an intra-chunk transformer block and an inter-chunk transformer block with an appropriate permutation of the input tensors, as shown in Fig. 3. The intra-chunk block models the short-time relationships, whereas the inter-chunk block

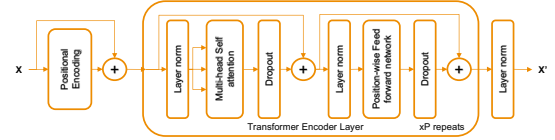


Figure 4: Transformer block.

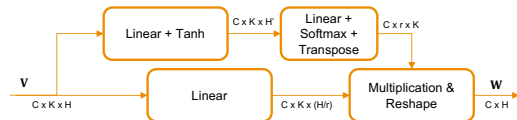


Figure 5: Sequence aggregation scheme.

models the long-time relationships in the input. Both intra- and inter-chunk blocks consist of a stack of transformer layers as shown in Fig. 4. The transformer encoder uses the dot-product self-attention [20]. In the present work, we use 4 transformer layers in the intra-chunk block and 2 transformer layers in the inter-chunk block. The output \mathbf{V} of the dual-path block is then fed to the attractor generation block shown in the lower section of Fig. 2

Attractor generation using EDA: The attractor generation block first computes an aggregate representation (\mathbf{W}) for all chunks, which is then fed to the EDA block to generate the attractors. For intra-chunk sequence aggregation, we consider the weighted averaging scheme shown in Fig. 5. The aggregated representation \mathbf{W} is a concatenation of $r = 4$ different weighted combinations of a lower-dimensional projection of the input \mathbf{V} using learnable weights, as shown in the top section of Fig. 5. The weights are learned in $H' = 2H$ dimensional space. This approach allows to learn to represent the time regions with different dominant speakers in different sub-spaces.

The operation of the EDA module is shown in Fig. 6 and is similar to the approach described in [19]. The input to the EDA is fed to a uni-directional LSTM [21] encoder with H units. The state of the LSTM encoder $\mathbf{c}_{e,-1}$ is initialized with a vector of zeros. The attractors are required to represent the speakers in the recording, and they need not model the time sequence of chunks. To promote learning the speaker characteristics, the input is shuffled randomly along the chunk-index dimension during training. The encoder state at the last input step $\mathbf{C}_{e,C}$ is considered as the recording level summary vector and used as the initial state of the LSTM decoder $\mathbf{c}_{d,-1}$. The decoder takes zero vectors as input and generates the attractors at the output.

A total of $(J + 1)$ number of attractors are generated for a recording with J number of sources. During training, J is assumed to be known and the $(J + 1)$ attractors are fed to a linear-sigmoid layer which is trained to predict a J length sequence of

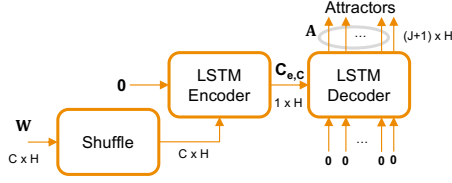


Figure 6: Encoder-decoder attractor (EDA).

Algorithm 1 EDA processing during inference

Input: Sequence Aggregation output \mathbf{W}
 $\sim, \mathbf{c}_{e,C} = \text{LSTM encoder}(\mathbf{W}, \mathbf{0})$
 Attractor matrix $\mathbf{A} = []$
 $j = 0$
 $\mathbf{a}_j, \mathbf{c}_{d,j} = \text{LSTM decoder}(\mathbf{0}, \mathbf{c}_{e,C})$
while Linear-Sigmoid(\mathbf{a}_j) ≥ 0.5 **do**
 $\mathbf{A} = \text{Concatenation}(\mathbf{A}, \mathbf{a}_j)$
 $j \leftarrow j + 1$
 $\mathbf{a}_j, \mathbf{c}_{d,j} = \text{LSTM decoder}(\mathbf{0}, \mathbf{c}_{d,j-1})$
end while
 $J = \text{length}(\mathbf{A})$
 return \mathbf{A}, J

1s followed by a 0. During inference, the linear-sigmoid layer output is used to infer J as described in Algorithm 1. The first J attractors (\mathbf{A}) corresponding to the sources are combined with the dual-path block output (\mathbf{V}) by element-wise multiplication to generate J channel 4D output (\mathbf{Y}). \mathbf{Y} is then input to the triple-path block shown in Fig. 7.

Triple-Path Block: The triple-path block extends the dual-path block (Fig. 3) with an additional inter-channel transformer block, as shown in Fig. 7. The inter-channel block models the relationships across the channels at all time steps. In the triple-path block, the channel dimension is treated the same as the batch dimension to apply the intra-chunk and inter-chunk transformer blocks. In contrast, the chunk and time dimensions are collapsed to the batch dimension to apply the inter-channel transformer block. We use a stack of 2 transformer layers in the inter-channel transformer block.

Mask prediction: The triple-path block output \mathbf{Z} is passed through a pReLU layer before conducting the overlap-add (OVA) operation. The OVA output is then passed through a gated output layer comprising two linear layers and is fed to a linear layer with ReLU activation at the output to generate the final masks $\mathbf{m}_j, j \in \{1, 2, \dots, J\}$. The output processing after the pReLU layer is similar to the SepFormer [14].

2.4. Loss Function

The loss function used is the sum of the source separation loss (\mathcal{L}_{sep}) and the speaker counting loss (\mathcal{L}_{spk}), i.e.,

$$\mathcal{L} = \mathcal{L}_{sep} + \mathcal{L}_{spk}. \quad (1)$$

For \mathcal{L}_{sep} , we use the negative of the scale-invariant signal-to-noise ratio (SI-SNR) loss, computed with optimal permutation of targets and their estimates, which has been used successfully for source separation [10, 14]. For a signal \mathbf{s} and its estimate $\hat{\mathbf{s}}$, the SI-SNR is computed as

$$\text{SI-SNR} = 10 \log_{10} \frac{\|\mathbf{s}^*\|^2}{\|\mathbf{s}^* - \hat{\mathbf{s}}\|^2} \text{ dB}, \quad (2)$$

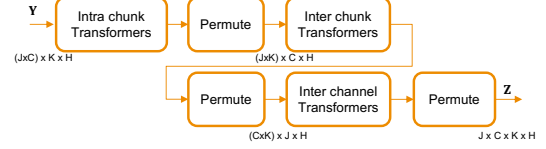


Figure 7: Triple-path block.

where $\mathbf{s}^* = \frac{\langle \hat{\mathbf{s}}, \mathbf{s} \rangle}{\langle \hat{\mathbf{s}}, \hat{\mathbf{s}} \rangle} \mathbf{s}$ is the scaled reference signal. The speaker counting head has binary outputs and hence the binary cross-entropy loss is used as \mathcal{L}_{spkr} .

3. Experiments

3.1. Dataset

We use the WSJ0-mix dataset [22] for the experimentation in this paper, which is also the most commonly used dataset for speaker-independent single-channel source separation [10, 14, 22]. The dataset defines simulated speech mixtures composed using the clean speech signals from the WSJ dataset [23]. The set WSJ0- J mix contains speech mixtures with J speakers, where $J \in \{2, 3, 4, 5\}$. The speaker signals are mixed with different relative levels chosen randomly in the range $[0 - 5]$ dB. The dataset comprises of 20K training examples, 5K validation examples, and 3K test examples for each of the 2 – 5 speaker mixture subsets. The training and test sets are composed using different sets of speakers. We consider the $f_s = 8$ kHz sampling rate version of the dataset for the experimentation.

3.2. Training Details

We trained the proposed architecture in three different ways:

- SepEDA₂: architecture trained on the two-speaker mixtures (WSJ0-2mix) alone. In this case, the speaker counting part of the network does not learn to count the speakers since the number of speakers is fixed for all training examples. However, the two attractors generated by EDA can still capture the speaker information at the recording level.
- SepEDA_[2-5]: architecture initialized with the trained SepEDA₂ model and fine-tuned using all the WSJ0-mix data, i.e., all sets with 2 – 5 speakers in the mixtures. The total number of training examples per epoch in this scenario is 80K.
- SepEDA_{2/3}: architecture trained on both the WSJ0-2mix and WSJ0-3mix datasets, i.e., 40K training examples per epoch.

We trained the models using the SpeechBrain [24] platform. We used the Adam optimizer [25] with an initial learning rate of 1.5×10^{-4} and a batch size of 1. The learning rate was kept fixed for the first 85 epochs and halved later if the validation SI-SNR did not decrease for two consecutive epochs. For the fine-tuning in SepEDA_[2-5], the initial learning rate was chosen as 1.5×10^{-5} and kept fixed for the first 10 epochs. A threshold of -30 dB was applied to the SI-SNR loss and the norm of the gradients was clipped to 5 during training. Time-domain 3-way speed perturbation [26] with a factor of 5% was also applied to the training examples. The SepEDA₂ and SepEDA_{2/3} models were trained for 200 epochs and the SepEDA_[2-5] model was trained for 50 epochs. The weights at the epoch corresponding to the best validation performance were used for the final evaluation. Illustrative audio examples

Table 1: Results for the WSJ0-2mix dataset.

| Architecture | #Parameters | SI-SNRi | SDRi |
|---------------------|-------------|-------------|-------------|
| Dual-Path RNN [10] | 2.6M | 18.8 | 19.0 |
| SepFormer [14] | 26M | 20.4 | 20.5 |
| Wavesplit [15] | 29M | 21.0 | 21.2 |
| SepEDA ₂ | 12.5M | 21.2 | 21.4 |

are available at <https://www.audiolabs-erlangen.de/resources/2022-Interspeech-BSS-EDA>

3.3. Performance Measures

We study the source separation performance using the SI-SNR improvement (SI-SNRi) and the source-to-distortion ratio (SDR) [27] improvement (SDRi) measures. We consider two cases, (i) the number of sources is known, and (ii) the number of sources is estimated. In the latter scenario, the estimated number of sources need not match the ground truth. When the number of sources is over-estimated, the subset of estimated sources matching best with the target sources are used for the performance measure computation. When the number of sources is under-estimated, a silence signal (i.e., an all-zeros signal) is used as the estimate for the sources not estimated.

3.4. Results

First, we studied the results on the WSJ0-2mix, for the SepEDA₂ model. Table 1 shows the source separation results compared with three different approaches in the literature. For fairness, we show the performance without the dynamic mixing data augmentation for all the approaches. SI-SNRi obtained is better than the current SOTA for the SepEDA₂ model. We note that the proposed architecture differs from the SepFormer only in the attractor generation and the triple-path processing. The attractors can be interpreted as global recording-level representations of speakers in the recording, which help in conditioning the subsequent blocks to separate the sources. The J channels of the triple-path block input can be interpreted as J source channels and the inter-channel block resolves the source permutations across these J channels in different chunks, i.e., it solves the problem of a given speaker appearing at different output channels in different chunks. Overall, the proposed modifications improve the performance over the SepFormer with a smaller size architecture.

Next, we show the performance of the SepEDA_[2-5] model, evaluated on the 2-5 mixture sets of WSJ0-mix dataset. Table 2 shows the performance comparison, in terms of SI-SNRi. We see that the proposed approach outperforms the RecursiveSS [16] and MulCAT [17] approaches in both the evaluations with known and estimated number of sources. The degradation when the number of sources is estimated is less for the 2 – 4 speaker mixtures. The source counting results are shown in Table 3. The source counting accuracy is found to be more than 90% with the proposed approach.

More often, practical conversations have no more than 3 concurrent speakers. To study this case, we considered the SepEDA_{2/3} model. Table 4 shows the results obtained for the WSJ0-2mix and WSJ0-3mix test sets. The performance of SepEDA_{2/3} is better than SepEDA₂ on the WSJ0-2mix, shown in Table 1. For comparison, we also show the results for SepFormer [14] and Wavesplit [15] architectures, which are trained separately on WSJ0-2mix and WSJ0-3mix datasets, and the number of speakers is known during evaluation. SepEDA_{2/3} model has significantly better SI-SNRi than the current SOTA.

Table 2: SI-SNRi (dB) performance comparison of three networks trained on WSJ0-mix. (**) denotes the evaluation scenario with known number of sources)

| | 2 | 3 | 4 | 5 |
|---------------------------|-------------|-------------|-------------|-------------|
| Recursive SS* [16] | 14.8 | 12.6 | 10.2 | - |
| MulCAT* [17] | 20.1 | 16.9 | 12.9 | 10.6 |
| MulCAT [17] | 18.6 | 14.6 | 11.5 | 10.4 |
| SepEDA _[2-5] * | 21.1 | 18.6 | 14.7 | 12.1 |
| SepEDA _[2-5] | 21.1 | 18.4 | 14.4 | 11.6 |

Table 3: Source counting results for the SepEDA_[2-5] model.

| Number of Sources | Estimated Number of Sources | | | |
|-------------------|-----------------------------|-------|--------|--------|
| | 2 | 3 | 4 | 5 |
| 2 | 99.8% | 0.2% | 0 | 0 |
| 3 | 0.47% | 97.0% | 2.53% | 0 |
| 4 | 0.03% | 1.9% | 90.17% | 7.9% |
| 5 | 0 | 0 | 3.13% | 96.87% |

Table 4: Results for the WSJ0-(2,3) mix datasets. (**) denotes the evaluation scenario with known number of sources). SDRi is computed only for recordings with correct source number estimation.

| Architecture | #Parameters | J | SI-SNRi | SDRi |
|-------------------------|-------------|---|-------------|-------------|
| SepFormer* [14] | 26M | 2 | 20.4 | 20.5 |
| | | 3 | 17.6 | 17.9 |
| WaveSplit* [15] | 29M | 2 | 21.0 | 21.2 |
| | | 3 | 17.3 | 17.6 |
| SepEDA _{2/3} * | 12.5M | 2 | 21.5 | 21.7 |
| | | 3 | 19.9 | 20.1 |
| SepEDA _{2/3} | 12.5M | 2 | 21.5 | 21.7 |
| | | 3 | 19.7 | 20.2 |

The performance is better even when the number of sources is estimated automatically. We see that the performance is also better than SepEDA_[2-5], shown in Table 2, for WSJ0-2mix and WSJ0-3mix. The speaker counting results showed that only one out of the 3K examples is wrongly identified as having three speakers for the WSJ0-2mix, and 23 examples are wrongly identified as having two speakers for the WSJ0-3mix.

4. Conclusions

In this paper, we propose a unified architecture for separating an unknown number of speech sources, combining encoder-decoder-based attractor calculation with dual-path transformer processing for long sequence modeling. We trained the proposed architecture on the WSJ0-mix dataset and showed that a better separation performance is obtained for speech mixtures with up to 5 speakers compared to the current best models for the same task. Future work would investigate the model performance on reverberant, noisy speech mixtures and long conversations with partially overlapping speakers.

5. Acknowledgment

The authors thank the Erlangen Regional Computing Center (RRZE) for providing computing resources and support.

6. References

- [1] P. Smaragdis, B. Raj, and M. Shashanka, "A probabilistic latent variable model for acoustic modeling," *Advances in models for acoustic processing, NIPS*, vol. 148, pp. 8–1, 2006.
- [2] M. N. Schmidt and R. K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," in *Inter-speech*, vol. 2, 2006, pp. 2–5.
- [3] G. Hu and D. Wang, "A tandem algorithm for pitch estimation and voiced speech segregation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 8, pp. 2067–2079, 2010.
- [4] K. Hu and D. Wang, "An unsupervised approach to cochannel speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 1, pp. 122–131, 2013.
- [5] R. J. Weiss and D. P. Ellis, "Speech separation using speaker-adapted eigenvoice speech models," *Computer Speech & Language*, vol. 24, no. 1, pp. 16–29, 2010, speech Separation and Recognition Challenge. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S088523080800017X>
- [6] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, pp. 2136–2147, 2015.
- [7] Y. Z. Isik, J. L. Roux, Z. Chen, S. Watanabe, and J. R. Hershey, "Single-channel multi-speaker separation using deep clustering," in *Interspeech*, 2016.
- [8] Y. Luo and N. Mesgarani, "Tasnet: Time-domain audio separation network for real-time, single-channel speech separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 696–700.
- [9] —, "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.
- [10] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-Path RNN: Efficient long sequence modeling for time-domain single-channel speech separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 46–50.
- [11] J. Chen, Q. Mao, and D. Liu, "Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation," in *Interspeech*, 2020, pp. 2642–2646.
- [12] Z. Zhang, B. He, and Z. Zhang, "TransMask: A compact and fast speech separation model based on transformer," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5764–5768.
- [13] C. Li, Y. Luo, C. Han, J. Li, T. Yoshioka, T. Zhou, M. Delcroix, K. Kinoshita, C. Boeddeker, Y. Qian, S. Watanabe, and Z. Chen, "Dual-path RNN for long recording speech separation," in *IEEE Spoken Language Technology Workshop (SLT)*, 2021, pp. 865–872.
- [14] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.
- [15] N. Zeghidour and D. Grangier, "Wavesplit: End-to-end speech separation by speaker clustering," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2840–2849, 2021.
- [16] N. Takahashi, S. Parthasarathy, N. Goswami, and Y. Mitsufuji, "Recursive speech separation for unknown number of speakers," in *Interspeech*, 2019, pp. 1348–1352.
- [17] E. Nachmani, Y. Adi, and L. Wolf, "Voice separation with an unknown number of multiple speakers," in *International Conference on Machine Learning (ICML)*. PMLR, 2020, pp. 7164–7175.
- [18] J. Zhu, R. A. Yeh, and M. Hasegawa-Johnson, "Multi-decoder DPRNN: Source separation for variable number of speakers," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 3420–3424.
- [19] S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, and K. Nagamatsu, "End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors," in *Interspeech*, 2020, pp. 269–273.
- [20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6000–6010.
- [21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [22] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 31–35.
- [23] J. S. Garofolo *et al.*, "CSR-I (WSJ0) Complete LDC93S6A," 1993.
- [24] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "SpeechBrain: A general-purpose speech toolkit," 2021, arXiv:2106.04624.
- [25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (Poster)*, 2015. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [26] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Interspeech*, 2015.
- [27] 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, 2006.