Speaker-conditioned Target Speaker Extraction Based on Customized LSTM Cells

Ragini Sinha*, Marvin Tammen†, Christian Rollwage*, Simon Doclo*,†

* Fraunhofer Institute for Digital Media Technology, Project group Hearing, Speech and Audio Technology, Oldenburg, Germany

† Department of Medical Physics and Acoustics and Cluster of Excellence Hearing4all, University of Oldenburg, Germany

Email:ragini.sinha@idmt.fraunhofer.de, marvin.tammen@uni-oldenburg.de, christian.rollwage@idmt.fraunhofer.de, simon.doclo@uni-oldenburg.de

Abstract

Speaker-conditioned target speaker extraction systems rely on auxiliary information about the target speaker to extract the target speaker signal from a mixture of multiple speakers. Typically, a deep neural network is applied to isolate the relevant target speaker characteristics. In this paper, we focus on a single-channel target speaker extraction system based on a CNN-LSTM separator network and a speaker embedder network requiring reference speech of the target speaker. In the LSTM layer of the separator network, we propose to customize the LSTM cells in order to only remember the specific voice patterns corresponding to the target speaker by modifying the information processing in the forget gate. Experimental results for two-speaker mixtures using the Librispeech dataset show that this customization significantly improves the target speaker extraction performance compared to using standard LSTM cells.

1 Introduction

Recently, the problem of speaker extraction has attracted a lot of attention in the speech processing community. The goal of speaker extraction is to extract a target speaker signal from a mixture of multiple speakers. In principle, this can be achieved by first extracting all individual speakers from the mixture using blind source separation [1–6] and then selecting the extracted signal that corresponds to the target speaker. However, a disadvantage of such a two-step approach is that the number of speakers needs to be known or estimated, which is not trivial in practice. Alternatively, one-step approaches have been proposed which aim at directly extracting the target speaker from the mixture by utilizing auxiliary information about the target speaker [7–16]. Such approaches are also referred as speaker-conditioned target speaker extraction techniques. Commonly used auxiliary information includes reference speech of the target speaker [7–11], video of the target speaker [12–14, 17], directional information [15, 18, 19], or information about the speech activity of the target speaker [16]. In this paper, we focus on single-channel target speaker extraction using reference speech as auxiliary information.

The single-channel speaker-conditioned target speaker extraction systems in [8–11] consist of two networks: a separator network and a speaker embedder network. The speaker embedder network is used to generate embeddings from the reference speech of the target speaker. These embeddings are used along with the mixture in the separator network with the goal to extract the target speaker signal from the mixture. The separator networks in [8, 9] estimate a time-frequency mask to perform speaker extraction, whereas the separator networks in [10, 11] perform speaker extraction in the time-domain. The embedder networks in [8, 11] utilize the same LSTM-based architecture of a speaker verification system proposed in [20], whereas different ResNet-based speaker verification systems are utilized in [9, 10] to generate embeddings of the target speaker. In [10] the speaker embedder network and the separator network are trained jointly using a weighted combination of a cross-entropy loss function and a scale-invariant signal-to-noise ratio (SI-SNR) loss function, while in [8, 9, 11] the speaker embedder network and the separator network are trained separately. The system in [8] utilizes a convolutional long short-term memory (CNN-LSTM) separator network trained with a power-law compression loss function, while the system in [9] utilizes an attention-based separator network trained with a squared l2-norm loss function and the system in [11] utilizes a separator network similar to [5] trained with an SI-SNR loss function. In this paper, our proposed speaker-conditioned target speaker extraction system is mostly inspired by the system presented in [8] (VoiceFilter).

In order to improve the performance of the baseline system in [8], in this paper we propose two modifications. First, instead of using standard LSTM cells in the separator network we introduce an LSTM cell that is customized for speaker-conditioned target speaker extraction. The purpose of this customization is to force the forget gate of the LSTM cell to only remember the specific voice patterns corresponding to the target speaker, which are needed to distinguish between the target speaker and the other speakers present in the mixture. Second, instead of using the spectral-domain power-law compression loss function we use the time-domain SI-SNR loss function. Experimental results for two-speaker mixtures from the Librispeech dataset show that the target speaker extraction performance can be significantly improved compared to the baseline system [8] in terms of signal-to-distortion ratio (SDR) [21] and perceptual evaluation of speech quality (PESQ) [22].

The remainder of this paper is organized as follows. In Section 2 we introduce the considered target speaker extraction system, where we mainly focus on the proposed customization of the LSTM cells. Section 3 discusses the network architectures and the parameters used for training and evaluation. Section 4 presents the target speaker extraction results.

2 Target Speaker Extraction System

We consider a scenario where a single microphone records a mixture of I speakers, i.e., in the time-domain the micro-

phone signal can be written as

$$y(n) = \sum_{i=1}^{I} x_i(n), \qquad (1)$$

where $x_i(n)$ denotes the speech signal of the i-th speaker and n denotes the discrete-time index. The goal is to extract the speech signal corresponding to the target speaker from the mixture, where without loss of generality the j-th speaker will be assumed to be the target speaker.

Similarly to [8–11], we consider a speaker-conditioned target speaker extraction system consisting of two parts (see Figure 1), namely, a speaker embedder network and a separator network. The speaker embedder network is used to generate the embeddings from the reference speech of the target speaker. Based on the target speaker embeddings and the mixture, the separator network aims at extracting the target speaker signal from the mixture. In this section, we will mainly discuss the separator network, while details about the embedder network will be provided in Section 3.1.

In [8, 9] speaker extraction was performed in the short-time Fourier transform (STFT) domain. The magnitude spectrum of the target speaker signal was estimated from the magnitude spectrum of the microphone signal using a (soft) mask $M(k,l)$, i.e.,

$$\tilde{X}_j(k,l) = M(k,l)Y(k,l), \qquad (2)$$

where k and l denote the frequency index and the time frame index, respectively. Using the CNN-LSTM network proposed in [8], the mask was computed from the target speaker embeddings $\mathbf{e}_j$ and the mixture as

$$M(k,l) = \phi(\mathbf{r}, \mathbf{e}_j), \qquad (3)$$

$$\mathbf{r} = \mathbf{g}(|Y(k,l)|), \qquad (4)$$

where $\mathbf{g}(\circ)$ denotes the convolutional layers of the separator network used to obtain the intermediate representation $\mathbf{r}$, and $\phi(\circ)$ denotes the rest of the separator network.

Instead of using standard LSTM cells in $\phi(\circ)$, we propose to customize the information processing through the forget gate of the LSTM cells such that the specific voice patterns of only the target speaker are remembered, which helps to distinguish the target speaker from the other speakers in the mixture when performing the extraction. After briefly reviewing the standard LSTM cell used in the baseline system [8] in Section 2.1, in Section 2.2 we discuss the proposed customization of the LSTM cell.

2.1 Standard LSTM cell

The working principle of an LSTM cell [23, 24] is determined by its state and three gates: the forget gate, the input gate, and the output gate (see Figure 2). The cell state behaves like the memory of the network with the ability to retain information through time, while the gates can add or remove information at each step t. In the following, the weight matrix and the bias of the forget gate, the input gate, the output gate and the control update are denoted by $\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_c, \mathbf{W}_o$ and $\mathbf{b}_f, \mathbf{b}_i, \mathbf{b}_c, \mathbf{b}_o$, respectively. The current and previous cell states are denoted by $\mathbf{c}_t$ and $\mathbf{c}_{t-1}$, while the current and previous hidden states are denoted by $\mathbf{h}_t, \mathbf{h}_{t-1}$.

As can be seen from (3), the input to the LSTM layer (and hence each LSTM cell) is the concatenation of the
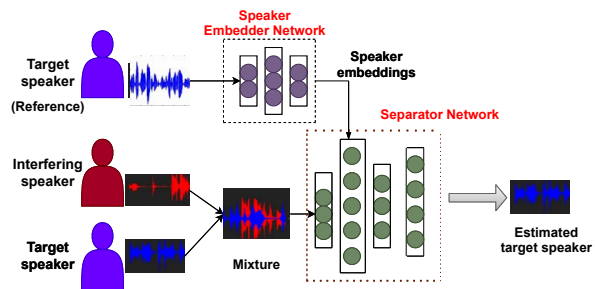

Figure 1: Block diagram of the used speaker-conditioned target speaker extraction system, consisting of a speaker embedder network and a separator network.

speaker embeddings $\mathbf{e}_j$ and the output of the convolutional layers $\mathbf{r}$. The recursive nature allows the LSTM cell to store information from the previous state. The forget gate of the LSTM cell decides which information should be retained or disregarded based on the previous hidden state and the current input. The information is retained in the cell state if the output of the forget gate is close to 1, otherwise it is disregarded. The output of the forget gate is obtained as

$$\mathbf{f}_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, (\mathbf{r}, \mathbf{e}_j)] + \mathbf{b}_f), \qquad (5)$$

where σ denotes the Sigmoid activation function.

The input gate decides which information is updated and stored in the cell state, and its output is obtained as

$$\mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, (\mathbf{r}, \mathbf{e}_j)] + \mathbf{b}_i). \qquad (6)$$

The cell state behaves like the memory of the network, and is updated as

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c[\mathbf{h}_{t-1}, (\mathbf{r}, \mathbf{e}_j)] + \mathbf{b}_c), \qquad (7)$$

$$\mathbf{c}_t = \mathbf{f}_t * \mathbf{c}_{t-1} + \mathbf{i}_t * \tilde{\mathbf{c}}_t, \qquad (8)$$

where $*$ denotes point-wise multiplication. Finally, the output gate decides which part of the cell state is transferred to the next hidden state, and its output is obtained as

$$\mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, (\mathbf{r}, \mathbf{e}_j)] + \mathbf{b}_o). \qquad (9)$$

Finally, the hidden state is updated as

$$\mathbf{h}_t = \mathbf{o}_t * \tanh(\mathbf{c}_t). \qquad (10)$$

2.2 Customized LSTM cell

As already mentioned, the forget gate is used to retain the relevant information and disregard irrelevant information, based on the previous hidden state and the current input. With the specific goal of speaker extraction in mind, intuitively the LSTM cell is supposed to learn to retain information related to the target speaker, while disregarding information unrelated to the target speaker, i.e., originating from the other speakers present in the mixture.

However, since in practice this will not be perfectly achieved, we propose to customize the LSTM cell in order to only retain the target speaker information by changing the information processing through the forget gate (see Figure 3). Instead of considering the concatenation of the
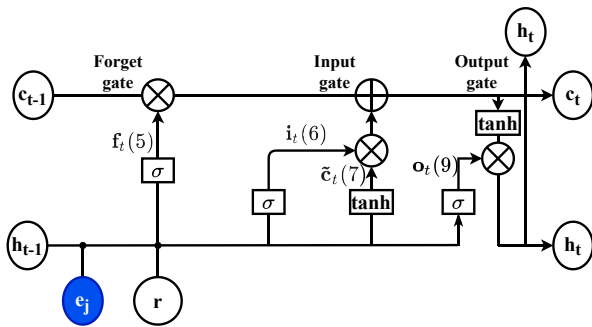
Figure 2: Standard LSTM cell.

Figure 3: Proposed customized LSTM cell for speaker-conditioned target speaker extraction.

target speaker embeddings $\mathbf{e}_j$ and the output of the convolutional layer $\mathbf{r}$, we only consider the target speaker embeddings, i.e.,

$$\mathbf{f}_t = \sigma(\mathbf{W}_e[\mathbf{h}_{t-1}, \mathbf{e}_j] + \mathbf{b}_e), \tag{11}$$

where $\mathbf{W}_e$ and $\mathbf{b}_e$ denote the weight matrix and the bias of the customized forget gate. It should be noted that all other gates, i.e., the input and output gates, and the cell update remain the same as described in the previous section.

In the customized LSTM cell, the forget gate in (11) aims at mapping the target speaker close to 1. This allows the current cell state in (8) to retain the target speaker information by multiplying the previous cell state with a value close to 1, while disregarding the information related to the other speakers from the previous cell state $\mathbf{c}_{t-1}$. We only customize the forget gate, since the forget gate is the main gate modifying the cell state. We do not customize the input and output gates. Since the input gate can only add but cannot remove information from the current cell state, a similar customization for the input gate might not be as effective as for the forget gate. A similar customization for the output gate may even lead to the loss of relevant information in the next hidden state.

3 Experimental Setup

In this section, we discuss the network architecture, the used parameters and the training procedure for the speaker embedder network and the separator network.

3.1 Speaker Embedder Network

We have used the same speaker embedder network as the baseline system [8], namely the speaker verification network originally proposed in [20]. It consists of 3 LSTM layers, each having 768 nodes. As input features the network uses 40-dimensional log-Mel-features, which are computed using an FFT size of 512 at a sampling frequency of 16 kHz, a Hann window with a frame length of 400 and a frame shift of 160 samples. Given the log-Mel-features of the reference speech of a target speaker, the speaker embedder network generates fixed (time-independent) 256-dimensional embeddings to be utilized further in the separator network.

We have used the official training and validation split of the Voxceleb dataset [25] to retrain the speaker embedder network. The Voxceleb dataset is a large speaker verification dataset that consists of more than one million utterances from more than 7000 speakers.
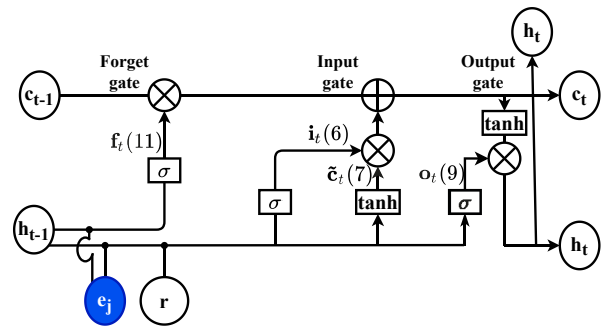
3.2 Separator Network

Similarly as for the baseline system [8], the separator network consists of eight 2D dilated convolutional layers, a customized LSTM layer and two fully connected (FC) layers. Each convolutional layer is followed with a batch-normalization layer and a ReLU activation function, where the dilated convolutional layers are used to increase the size of the receptive field of the network. The number of filters, filter-size and the dilation parameters for the convolutional layers are the same as for the baseline system [8] (see Table 1). The only difference in the architecture of the proposed separator network compared to the baseline is the number of nodes for the customized LSTM layer and the FC layers. We have used 600 nodes for the customized LSTM layer and the first FC layer consists of 514 nodes followed with a ReLU activation function, while the second FC layer consists of 257 nodes with a Sigmoid activation function. As input features the network uses the magnitude of the STFT coefficients of the mixture. The STFT coefficients are computed using an FFT size of 512 at a sampling frequency of 16 kHz, a square-root Hann window with a frame length of 512 and a frame shift of 256 samples.

To train the separator network we have considered two different loss functions: a spectral-domain loss function based on power-law compression (PLC) [8], and the scale-invariant signal-to-noise ratio (SI-SNR) loss function in the time-domain [5]. In total we have trained three different separator networks:

1. **Standard / PLC**: using the PLC loss function and the standard LSTM cells. This separator network is similar to the baseline system in [8]. Since the baseline system is not publicly available, we retrained it using the same loss function and model parameters.

2. **Standard / SI-SNR**: using the SI-SNR loss function and standard LSTM cells.

3. **Customized / SI-SNR**: using the SI-SNR loss function and the proposed customized LSTM cells.

The Adam optimizer [26] with a learning rate of 0.0002 was used. We used a batch size of 16 and fixed the total number of epochs to 50, while clipping the gradient norm to 10. An early stopping criterion was used to prevent the network from further training if the validation loss did not decrease after 7 epochs.

To train and validate our separator networks, we have used the official training and validation set from the Librispeech dataset [27]. To generate the training and validation triplets (mixture, reference and target), we have used 100 hours of clean speech of training set and clean speech

Layer	Kernel Size	Dilation	Filters /Nodes	Baseline [8]
Conv1	(1×7)	(1×1)	64	64
Conv2	(7×1)	(1×1)	64	64
Conv3	(5×5)	$(2^0 \times 1)$	64	64
Conv4	(5×5)	$(2^1 \times 1)$	64	64
Conv5	(5×5)	$(2^2 \times 1)$	64	64
Conv6	(5×5)	$(2^3 \times 1)$	64	64
Conv7	(5×5)	$(2^4 \times 1)$	64	64
Conv8	(1×1)	(1×1)	8	8
Customized LSTM	-	-	600	-
Standard LSTM	-	-	-	400
FC 1	-	-	514	600
FC 2	-	-	257	600

Table 1: Parameters of the proposed separator network and Baseline(VoiceFilter) [8].

	Mean ΔSDR (dB)	Mean ΔPESQ
Baseline (VoiceFilter) [8]	5.50	-
Standard / PLC	5.62	1.95
Standard / SI-SNR	6.99	2.01
Customized / SI-SNR	**7.96**	**2.07**

Table 2: Mean SDR improvement (ΔSDR) and mean PESQ improvement (ΔPESQ) of baseline system (Voice-Filter), retrained system using standard LSTM cells and PLC loss function, system using standard LSTM cells and SI-SNR loss function, and proposed system using customized LSTM cells and SI-SNR loss function.

PESQ is improved by 0.06. Third, by using the proposed customized LSTM cells (with the SI-SNR loss function) the mean SDR can be further improved by 0.97 dB, while the mean PESQ can be further improved by 0.06. These results show that both proposed modifications significantly improve the performance compared to the baseline target speaker extraction system.

5 Conclusion

For a CNN-LSTM target speaker extraction system, in this paper we have proposed customized LSTM cells aimed at retaining information of the target speaker and disregarding irrelevant information of the other speakers. To this end, we have modified the forget gate of the LSTM cell, considering only the target speaker embeddings, while keeping the other gates and the cell state update unchanged. Experimental results for two-speaker mixtures using the Librispeech dataset show that the proposed customization yields performance improvements in terms of both SDR and PESQ compared to using standard LSTM cells. In future work, we will investigate the usage of the proposed customized LSTM cells when jointly training the speaker embedder network and the separator network, and we will consider different auxiliary modalities of the target speaker.

development set from the Librispeech dataset. We have followed the same procedure as in [8] to create the mixture, the reference speech and the target speech, each having a length of 4 seconds. The mixture is constructed (assuming I=2 speakers) by summing the utterances of two randomly chosen different speakers (target and interfering) from the dataset, while the reference speech is a randomly chosen utterance of the target speaker, which is completely different from the utterances used for constructing the mixture.

4 Results and Discussion

We have evaluated the performance of all considered target speaker extraction systems using the official test set of the Librispeech dataset [1]. The same procedure as for training has been used to generate the triplets (mixture, reference and target) (see Section 3.2). As performance measures, we have used the signal to distortion ratio (SDR) [21] and the perceptual evaluation of speech quality (PESQ) [22] measure. SDR is a common measure to evaluate the performance of source separation systems, while PESQ is a speech quality metric commonly used for speech enhancement systems. For both performance measures, the clean target speech signal was used as the reference signal.

For the three considered target speaker extraction systems, Table 2 shows the mean SDR improvement (in dB) and the mean PESQ improvement with respect to the mixture. For reference, this table also shows the mean SDR improvement of the baseline (VoiceFilter) reported in [8]. We do not show the mean PESQ improvement for the baseline as no PESQ results were reported in [8]. For the mixture, the mean SDR is equal to 0.14 dB and the mean PESQ is equal to 1.37. First, it can be observed that the mean SDR improvement of our retrained system using standard LSTM cells and PLC loss function is very similar to the mean SDR improvement of the original VoiceFilter [8], showing the validity of retraining the networks. Second, it can be observed that by using the SI-SNR loss function instead of the PLC loss function (with standard LSTM cells), the mean SDR is improved by 1.37 dB, while the mean

---

[1]Audio Examples can be found at `https://github.com/SinhaRagi/ITG_Target-Speaker-Extraction/tree/main/Audio_Examples`

References

[1] S. Makino, *Audio Source Separation*. Springer, 2018.

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

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

[4] 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 Trans. on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1901–1913, 2017.

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

[6] J. Le Roux, G. Wichern, S. Watanabe, A. Sarroff, and J. R. Hershey, "Phasebook and friends: Leveraging discrete representations for source separation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 370–382, 2019.

[7] K. Žmolíková, M. Delcroix, K. Kinoshita, T. Ochiai, T. Nakatani, L. Burget, and J. Černocký, "Speakerbeam:

Speaker aware neural network for target speaker extraction in speech mixtures," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 800–814, 2019.

[8] Q. Wang, H. Muckenhirn, K. Wilson, P. Sridhar, Z. Wu, J. R. Hershey, R. A. Saurous, R. J. Weiss, Y. Jia, and I. L. Moreno, "VoiceFilter: Targeted Voice Separation by Speaker-Conditioned Spectrogram Masking," in *Proc. Interspeech*, (Graz, Austria), pp. 2728–2732, Sept. 2019.

[9] T. Li, Q. Lin, Y. Bao, and M. Li, "Atss-net: Target speaker separation via attention-based neural network," in *Proc. Interspeech*, (Shanghai, China), pp. 1411–1415, Oct. 2020.

[10] M. Ge, C. Xu, L. Wang, E. S. Chng, J. Dang, and H. Li, "Spex+: A complete time domain speaker extraction network," in *Proc. Interspeech*, (Shanghai, China), pp. 1406–1410, Oct. 2020.

[11] Z. Zhang, B. He, and Z. Zhang, "X-tasnet: Robust and accurate time-domain speaker extraction network," in *Proc. Interspeech*, (Shanghai, China), pp. 1421–1425, Oct. 2020.

[12] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein, "Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation," *ACM Trans. on Graphics*, vol. 37, no. 4, pp. 1–11, 2018.

[13] T. Afouras, J. S. Chung, and A. Zisserman, "The conversation: Deep audio-visual speech enhancement," in *Proc. Interspeech*, (Hyderabad, India), pp. 3244–3248, Sept. 2018.

[14] C. Li and Y. Qian, "Listen, watch and understand at the cocktail party: Audio-visual-contextual speech separation," in *Proc. InterSpeech*, (Shanghai, China), pp. 1426–1430, Oct. 2020.

[15] R. Gu, L. Chen, S.-X. Zhang, J. Zheng, Y. Xu, M. Yu, D. Su, Y. Zou, and D. Yu, "Neural spatial filter: Target speaker speech separation assisted with directional information.," in *Proc. Interspeech*, (Graz, Austria), pp. 4290–4294, Sept. 2019.

[16] M. Delcroix, K. Žmolíková, T. Ochiai, K. Kinoshita, and T. Nakatani, "Speaker activity driven neural speech extraction," *arXiv preprint arXiv:2101.05516*, 2021.

[17] D. Michelsanti, Z.-H. Tan, S.-X. Zhang, Y. Xu, M. Yu, D. Yu, and J. Jensen, "An overview of deep-learning-based audio-visual speech enhancement and separation," *arXiv preprint arXiv:2008.09586*, 2020.

[18] A. Brendel, T. Haubner, and W. Kellermann, "A unified probabilistic view on spatially informed source separation and extraction based on independent vector analysis," *IEEE Trans. on Signal Processing*, vol. 68, pp. 3545–3558, 2020.

[19] A. Aroudi and S. Braun, "DBNET: DOA-driven beamforming network for end-to-end farfield sound source separation," *arXiv preprint arXiv:2010.11566*, 2020.

[20] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Calgary, Canada), pp. 4879–4883, Apr. 2018.

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

[22] ITU-T, "Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs P.862," tech. rep., International Telecommunications Union (ITU-T) Recommendation, Feb. 2001.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[25] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *Proc. Interspeech*, (Stockholm, Sweden), pp. 2616–2620, Aug. 2017.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd International Conference for Learning Representations*, (San Diego, USA), pp. 01–15, July 2015.

[27] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Brisbane, Australia), pp. 5206–5210, Apr. 2015.