INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Deeply supervised curriculum learning for deep neural network-based sound source localization

Min-Sang Baek, Joon-Young Yang, and Joon-Hyuk Chang

Department of Electronic Engineering, Hanyang University, Seoul, Republic of Korea
{kng643, dreadbird, jchang}@hanyang.ac.kr

Abstract

Deep neural network (DNN) has made impressive progress in sound source localization (SSL) tasks with the hard *n*-hot labels that represent specific directions-of-arrivals (DOAs). However, recent study suggested soft DOA labels, considering the correlations between targets and nearby DOAs. In this study, to effectively train a DNN using soft labels, we propose deeply supervised curriculum learning (DSCL) by adopting the two techniques for the DNN, deep supervision (DS) and curriculum learning (CL). We train a DNN to solve SSL problems progressing from easier to harder, expecting the DNN would gradually reduce the angular region of the target DOAs. It is gained by various resolution soft targets for the different DNN layers to deeply supervise the DNN, while increasing the angular selectivity of the targets from the early to late stages of training by CL. Proposed method was verified on datasets with multi-speakers, and exceeded the hard-label methods with great improvements.

Index Terms: sound source localization, direction-of-arrival, deep neural network, curriculum learning, deep supervision

1. Introduction

Sound source localization (SSL) aims at estimating the direction-of-arrival (DOA) of each sound source based on the signals received by a microphone array. Although many traditional SSL approaches such as the generalized cross-correlation phase transform [1], steered response power with phase transform [2], and multiple signal classification (MUSIC) [3] methods, have been widely adopted, recent studies exploit deep neural networks (DNNs) to solve SSL problems [4–11]. Since DNNs are typically trained using a supervised deep learning method, the selection of target labels is crucial to train DNN-based SSL models to effectively estimate the DOAs of sound sources. Specifically, we focus on the DNN-based SSL approach that performs classification of DOAs within a closed set of DOA labels.

DOA target labels for training DNN-based SSL models can be broadly classified into two categories: *hard* [5, 12] and *soft* [6] labels. The SSL problem is firstly formulated as a DNN-based classification problem in [4], and the posterior probability of the target DOAs are used as a target to specify the target directions. In [5], *n*-hot vector, where target directions are assigned as one and others as zero was used as the hard label for multi-speaker localization. However, more recent studies [6–9] argued that hard labels do not fully consider the correlations between adjacent DOAs, which may not sufficiently reflect the physical behavior of microphone array signals particularly when the DOA classifier uses high-resolution DOA targets. To deal with this problem, He *et al.* [6] defined the soft DOA labels by a probabilistic method by utilizing Gaussian-like functions, whose mean values are the target azimuths and variances determine the angular selectivity or the uncertainty of the DOA estimates produced by the DNN. Such target labels are designed to naturally consider the correlations between adjacent directions and were shown to be effective for training DNNs for SSL.

In this study, to effectively train a DNN-based SSL model using the soft DOA labels, we propose deeply supervised curriculum learning (DSCL) method. First, inspired by the deep supervision (DS) [13–15] technique that supervises intermediate DNN layers for improved error propagation, we propose to deeply supervise multiple layers of DNN-based SSL model by assigning low- to high-resolution DOA labels from the lower to upper layers of the DNN. Second, we propose a curriculum learning (CL) [16] method for training the DNN-based SSL model, which gradually increases the angular selectivity of the soft DOA labels as the training progresses. Lastly, DS and CL are applied simultaneously to utilize the merits of both methods. Consequently, we expect the model to gradually learn to solve from easier to harder SSL problems during the training, particularly when adverse acoustic conditions are encountered. The effectiveness of the proposed method is validated on both synthetic and recorded datasets comprising single or multiple speakers, achieving significant improvements over the conventional SSL approaches.

2. DNN-based SSL Framework

2.1. Input feature representation

Suppose N speech sources are captured by a C-channel microphone array in a noisy and reverberant room environment. In the short-time Fourier transform (STFT) domain, the signal at the l-th frame can be expressed as follows:

$$Y_c(l,f) = \sum_{s \in n_l} X_{c,s}(l,f) + V_c(l,f), \tag{1}$$

where c and f denote the microphone channel and frequency bin index, respectively, and n_l is the set of active speakers in the l-th frame. Also, X and V represent the speech and noise components, respectively.

For the input feature of the DNN, we use the instantaneous relative transfer function (iRTF) [17] to encode the spatial feature. Indeed, iRTF, $Z_c(l,f)$, for the c-th microphone channel, is calculated as follows:

$$Z_c(l,f) = \frac{Y_c(l,f)}{Y_{c*}(l,f)}. \tag{2}$$

where the subscript $c*$ denotes the reference microphone channel. The real and imaginary components of the iRTFs of all

10.21437/Interspeech.2023-2451

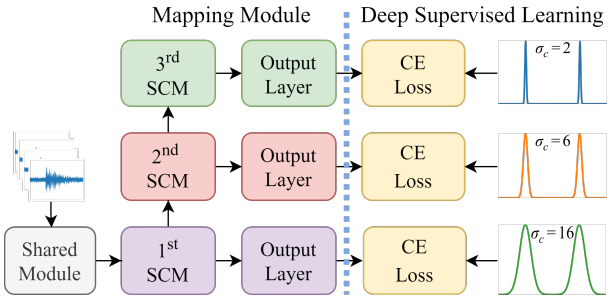

Figure 1: *CRNN that estimates $K = 3$ outputs. Each output layer is supervised with different SS to provide various properties of correlation and gradually decrease the ambiguity of the output of the deepest layer.*

channels, except for the reference channel, are stacked to form an input feature. The size of the input feature is $2(C-1) \times F$, where F denotes the number of frequency bins.

2.2. Model structure

Similar to [11, 18], we adopt the convolutional recurrent neural network (CRNN) architecture as the DNN-based SSL model. The CRNN consists of a shared module and a mapping module. The shared module comprises four causal convolution modules (CCMs) and three uni-directional gated recurrent units (GRUs) [19]. Each CCM performs causal 2D convolution with a 3×3 kernel, batch normalization (BN), exponential linear unit (ELU), and max pooling layer. The output of the last CCM is flattened and fed into the GRUs. Then, the output of the last GRU is introduced to the mapping module, which comprises K skip-connection modules (SCMs) and an output layer. An SCM performs 1D pointwise convolution, BN, and ELU, and the output layer comprises a 1D pointwise convolution followed by a sigmoid function to produce the posterior probabilities of DOAs. The output layer has $\frac{360}{r}$ units for DOA classification, where r denotes the angular resolution.

2.3. Target DOA labels

2.3.1. Hard label

Perhaps the most intuitive form of the target DOA labels for training a DNN-based SSL model is a hard n-hot vector, where 1's are assigned to the indices of the source locations [5]. By employing n-hot labels, the task becomes a typical multi-class classification problem, in which only the indices of the target directions according to a predetermined angular resolution r are of interest. However, as r increases, the DNN may suffer from dealing with SSL problems because the hard labels do not consider the relation between adjacent DOAs and assign 0's to the directions spatially close to the target direction.

2.3.2. Soft label

To address the aforementioned problems of hard labels, a soft labeling scheme was proposed in [6] and used in recent studies [7–9]. Specifically, to consider the correlations between the target and its adjacent directions, the target direction is assigned 1 and its adjacent directions are assigned the probabilistic values representing the source existence instead of 0's. For this reason, the soft DOA labels defined as such are sometimes referred to as spatial spectrum (SS) [6]. To produce the soft DOA labels or the target SS, we adopt the Gaussian-like function described in [6] as well as the spatial gain function described in [20].

$$g(l, \xi_c, \sigma_c) = \begin{cases} \max\limits_{\phi' \in \Phi_l} \{e^{\kappa(\xi_c, \sigma_c)(\cos(d(\phi, \phi'))-1)}\} & \text{if} |\Phi_l| > 0 \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

$$d(\theta, \theta') = \text{abs}(\theta - \theta'), \tag{4}$$

$$\kappa(\bar{\xi}, \bar{\sigma}) = \frac{\ln(\bar{\xi})}{\cos(\bar{\sigma}) - 1}, \tag{5}$$

where $g(l, \xi_c, \sigma_c)$ denotes the function for formulating target SS, $d(\theta, \theta')$ measures the absolute angular difference, and $\kappa(\bar{\xi}, \bar{\sigma})$ denotes the spatial gain function; ξ_c and σ_c are cutoff value and angle for soft label; and Φ_l denotes the set of azimuth angles of active speakers at l-th frame, with $|\cdot|$ representing the cardinality of the set. Following [20], ξ_c was set as $\frac{1}{\sqrt{2}}$. Examples of the SS set according to the different values of σ_c are illustrated on the rightmost panel of Fig. 1. Eq. (3) and [6, Eq. (3)] both assign similar value when the parameters are adjusted, respectively, but as Eq. (3) can assign ξ_c at σ_c, Eq. (3) makes synergy when adjusting the parameters to apply CL and DS described in latter section because assigned value could be known directly.

2.4. Iterative max-peak selection from predicted SS

DOAs are predicted by decoding the estimated SS. One widely using decoding method is max-peak selection [21], which was adopted by several studies [8, 11, 22]. When the number of active speakers $|n_l|$ is known, the indices for the highest peaks of the SS are considered as the estimated DOAs. To obtain the indices of the peaks, we adopt the iterative max-peak selection method described in [8], while neglecting the neighborhoods of the selected peaks within a predetermined angular distance margin, L. This can prevent the decoding algorithm from missing the active speakers whose posterior probabilities are represented by relatively small peak amplitudes. The iteration ends when the number of peaks reaches $|n_l|$.

3. Proposed Method

3.1. CL for DNN-based SSL

CL was first proposed by [16] to train a DNN effectively by increasing the difficulty of the training progressively. In the case of utilizing SS to train the DNN, SS with a wider σ_c can be regarded as an easier target because relatively larger values are assigned at the neighboring angles. A large σ_c indicates that there is a greater correlation between the target DOA and the neighboring directions, so the DNNs can consider such relations. On the other hand, a large σ_c can lead to ambiguity in the max-peak search for determining the DOAs from the estimated SSs. Because the output is not ideal, a few local-maximum points exist in the surroundings of the target directions, so it is difficult to distinguish the peak points from SS. In this case, SS with a smaller σ_c is more suitable for the max-peak search. Therefore, to take advantage of a large σ_c, but to minimize the drawbacks, we propose to use CL to define SS whose σ_c progressively decreases from a large to a small value.

At the beginning of training the DNN, we initialize σ_c with σ_{init}, which would be the largest value during entire training procedure. When each epoch ends, we subtract a specific positive value from the previous σ_c and fix the value when σ_c reaches σ_{end}. We anticipate that DNNs can take advantage of both easier and correlation-instructing targets at the beginning

of the training procedure and precisely DOA indicating target at the end. This method can create a synergistic effect when combined with DS.

3.2. DS for DNN-based SSL

DS was proposed to prevent the gradient vanishing problem by providing multiple targets for the outputs of several DNN layers [13]. In [15], DS was utilized using multi-resolution targets to produce low-resolution images at the lower layer and high-resolution images at the upper layer. Inspired by [15], we apply DS by supervising the DNN-based SSL model with SSs formed with different σ_c depending on the depth of the layers. This induces the model to produce an unambiguous SS suitable for the max-peak detection decoding method. In addition, the shared layers of a DNN can learn the general characteristics via multiple targets with different properties.

Among the K outputs of the CRNN, some outputs that are selected in advance form a set of outputs O that are used for training and inference, where o_k denotes an element in O and is the output of the k-th output layer. Among the predictions in O, the SS formulated with a relatively larger σ_c is given as the target for the output of the lower output layer. For the outputs from the upper layers, SSs produced with a comparatively smaller σ_c are provided. Each loss of each output layer is summed and backpropagated to update the parameters of the CRNN. In the inference stage, the predictions in O are independently utilized to estimate the DOAs. Moreover, CL and DS can be combined by different σ_{init}, σ_{end} and different decreasing schedulers for each layer, respectively.

To summarize the CL and DS methods, an easier target that indicates more correlation about adjacent directions is used to train the DNN at the beginning of training and the output of the lower layer. As training progresses, and for the output of the upper layer, a distinctive target is given to form more accurate SSs to distinguish the DOAs.

4. Experimental Setup

4.1. Dataset

In our experiments, we trained CRNN with a synthetic dataset and evaluated using both a synthetic develop dataset (SD) and a real-recorded LOCATA challenge dataset (RD) [23]. We used the NAO robot head array of [23], which was configured of 12 microphones. For the training set, we used the LibriSpeech corpus [24] for clean utterances, noise from the Microsoft Scalable Noisy Speech Dataset (MS-SNSD) [25], and randomly generated white noise. To generate the training samples, we followed the parameters of the room as [10]. The size of the room and the position of the array were randomly selected, but the position was at least 0.3 m far from the closest wall. In addition, the array was rotated freely on the xy-plane. We then selected N utterances from different speakers. We selected azimuths and elevations for the position of the speakers, but azimuth distance between adjacent speakers was at least 10°. In addition, one noise source was randomly selected from MS-SNSD. We then generated $N + 1$ room impulse responses (RIRs) for utterances and noise. To generate RIRs, we used gpuRIR toolkit [26]. From the beginning to the attenuation level of 12 dB, image source method [27] was used. Then, from the attenuation level of 12 dB to 40 dB, diffuse reverberation model was used to generate RIRs for diffuse sound. All utterances and noise source were convolved with each RIR and added together based on a randomly chosen signal-to-noise ratio (SNR) within the range we set, including white noise. To prevent overfitting and provide diversity in the data, training samples were generated on-the-fly.

Table 1: *Synthetic dataset generation parameters*

Parameter	Interval	Unit
SNR	5 – 30	dB
SNR between Utterances	-5 – 5	dB
RT60	0.2 – 1.3	s
Room Size	3×3×2.5 – 10×8×6	m^3
Absorption Weights of the Room	0.5 – 1.0	-
Height of the Array	0.3 – 1.2	m
Speaker-to-Array Distance	0.3 – 2.5	m
Azimuth	0 – 360	$^\circ$
Elevation	30 – 100	$^\circ$

For the SD, we used the LibriSpeech corpus test set, MS-SNSD test set, and white noise. The SD was generated with the same parameters as the training set, but with different speech and noise sources. Table 1 shows the parameters randomly sampled from uniform distribution within the interval for training dataset generation: SNR, SNR between utterances, RT60, room size, absorption weights of the room, height of the array, speaker-to-array distance, azimuth, and elevation candidates. For RD, we used LOCATA tasks 1 and 2, which were recorded in the real room environment with 0.55 s of RT60 and all speakers were static. From the task 2, which is configured for utterances of multiple speakers, we used the data that the maximum number of active speakers was 2 or less in the utterance.

4.2. Training specifications

The length of each sample of the training dataset and SD was 4 s. r was set as 1° and L was set as 10° considering the minimum azimuth distance between adjacent speakers. The sampling rate of the data was 16 kHz and the window length and hop length for STFT were 16 ms and 8 ms, respectively. We used the WebRTC voice activity detector [28] to obtain Φ_l and ϕ_l in the l-th frame. The model was trained for 50 epochs, with a batch size of 16. Cross-entropy (CE) loss was used for the loss function. We used the Adam optimizer [29] with a gradient clipped between ± 5. The learning rate started at 1e-3, and when the validation loss did not decrease, the learning rate decreased by the factor of 0.9. We evaluated the model that documented the lowest cross-validation loss during the training.

4.3. Evaluation metrics

The SSL performance was evaluated using two metrics: mean absolute error (MAE) and accuracy (ACC). MAE was measured as the angular distance between the oracle and the estimated azimuth angles using Eq. (4) with the unit of $^\circ$. For ACC, a frame was considered to be correctly classified if the angular distance between the oracle and the estimated azimuth angles was less than a predefined threshold with a unit of $\%$. The threshold was set as 10° in our experiments. Both MAE and ACC were computed for every frame, and the performances were compared based on the average of the results.

5. Results and Analysis

5.1. Performance with soft label

Table 2 shows the evaluation results of MUSIC [3], ResNet [9] and CRNNs trained on the soft labelled target with various σ_c. We modified ResNet with causal CNNs and trained it using SS. CRNNs were configured with $K = 1$ and the σ_c of the SS

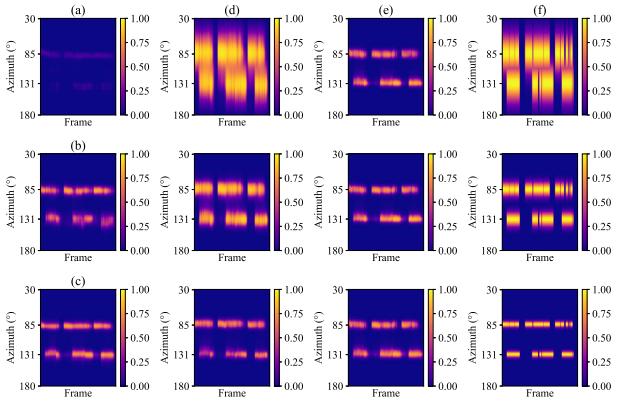

Figure 2: *Oracle SSs and estimates of real data, where speakers were located on* $85\,^\circ$ *and* $131\,^\circ$. *From (a) to (e), they are the outputs of the each model: (a), CRNN with* $\sigma_c = 0$*; (b), CRNN*$_1$*; (c), CRNN+CL*$_1$*; (d), CRNN+DS*^{d}*; and (e), CRNN+DSCL. For (d) and (e), each represents the output of* $k = 1, 2, 3$*, from the top. (f) is the oracle target with* $\sigma_c = 16, 6, 2.5$.

Table 2: *Performance of baselines and CRNNs trained on various* σ_c

Method	σ_c	Synthetic		LOCATA	
		MAE	ACC	MAE	ACC
MUSIC [3]	-	39.68	52.81	14.35	81.45
ResNet [9]	0	29.80	71.76	15.84	80.13
	2.5	25.85	75.93	8.74	87.21
CRNN	0	10.36	90.57	8.83	86.98
	1	9.53	91.20	4.66	93.27
	2.5	**8.40**	**92.50**	**4.06**	**93.79**
	6	9.92	92.22	5.26	92.39
	16	18.56	87.20	6.14	89.92

was fixed during the entire training period. Compared to MUSIC and ResNet, we identified that CRNN was suitable for low-latency, causal systems compared to other methods. The SS with $\sigma_c = 0$ represents the hard label, and the results show that SSs with $\sigma_c > 0$ were better targets for both ResNet and CRNN. Additionally, as shown in Fig. 2, the peaks of (a) are not distinguishable compared to (b). However, excessive σ_c made the performance worse on SD, because as we mentioned in previous sections, large σ_c induced ambiguous property to the model. Therefore, appropriate σ_c, like 2.5, should be utilized for the parameter of soft labelled target.

5.2. Performance with CL and DS

Table 3 compares the results of the CRNNs trained with five different methods: CRNN trained with static σ_c using the k-th output (CRNN$_k$), CRNN with only CL using the k-th output (CRNN+CL$_k$), CRNN with only DS and equal σ_c (CRNN+DS^e), CRNN with only DS and different σ_c (CRNN+DS^d), and CRNN with DSCL (CRNN+DSCL). Comparing CRNN$_1$, CRNN$_2$ and CRNN$_3$, which were trained only with the output of the first, second, and third output layers with static $\sigma_c = 2.5$, the model with deeper layers showed similar MAE and ACC on SD, but worse MAE and ACC on RD.

For CRNN+CL$_k$, the models with different k values were trained independently. σ_c was first set to σ_{init} and decreased until epoch 25; then, σ_c was fixed with σ_{end}. CRNN+CL$_k$ showed an improved performance on SD compared with CRNN$_k$, and $k = 2$ showed the best results among them. As shown in Fig. 2(c), training with CL could produce a clearer output for max-peak detection.

Table 3: *Performance of CRNN trained with CL, DS and both*

Method	σ_{init}	σ_{end}	k	Synthetic		LOCATA	
				MAE	ACC	MAE	ACC
CRNN$_k$	2.5	2.5	1	8.40	92.50	4.06	93.79
	2.5	2.5	2	8.46	92.55	4.67	93.40
	2.5	2.5	3	8.36	92.50	4.91	92.68
+CL$_k$	16	2.5	1	7.29	93.80	3.39	94.24
	16	2.5	2	6.69	94.27	3.22	94.26
	16	2.5	3	7.08	94.05	3.87	93.84
+DS^e	2.5	2.5	1	7.12	94.03	4.45	92.17
	2.5	2.5	2	7.09	94.05	4.49	92.28
	2.5	2.5	3	7.15	94.00	4.55	92.16
+DS^d	16	16	1	14.21	90.89	4.34	92.99
	6	6	2	7.72	94.29	3.60	93.92
	2.5	2.5	3	**6.33**	**94.95**	3.53	93.73
+DSCL	16	2.5	1	6.93	94.18	2.96	**94.88**
	6	2.5	2	6.79	94.35	**2.94**	94.73
	2.5	2.5	3	6.82	94.33	2.97	94.70

Also, CRNN+DS^e and CRNN+DS^d were configured with $K = 3$ and $k = 1, 2, 3$, and σ_c values were unchanged during the training period. The losses computed between the outputs and targets were weighted, summed, and used to train the models. Until epoch 25, the ratio between the losses of each layer was 1.1:1.0:0.9, after then the ratio changed to 1:1:1. The performance of CRNN+DS^e was superior to that of CRNN$_k$ on SD, but not on RD. In the case of CRNN+DS^d, the results from first and second layers showed worse performances in terms of both MAE and ACC. However, the output from $k = 3$ exhibited the best performance on SD in our experiments. As can be seen in Fig. 2(d), the outputs of lower layers were ambiguous for max-peak detection because of a large σ_c; therefore, the performance decreased.

Next, CRNN+DSCL was configured with the same K as that of CRNN+DS^d. In addition, σ_{init} on each layer was different, but σ_{end} was the same, 2.5, and the manner of decreasing σ_c was the same as that of CRNN+CL$_k$. The performance on SD was slightly worse than that of CRNN+CL$_k$ and CRNN+DS^d, but the performance on RD was significantly improved, and the best performance was recorded on both MAE and ACC. Fig. 2(e) is the output of CRNN+DSCL and is the clearest output among all the estimates.

6. Conclusion

In this study, we proposed a new training method for DOA estimating DNNs adopting DS, CL and both by adjusting the parameters of the soft label. The CL changed the parameters of the target SS to progressively produce clearer outputs, and the DS provided diverse target SSs to the output layers of different hierarchies. This scheme allowed DNNs to be trained more effectively with various properties that each target provides without increasing the model size and computation power. Our experimental results demonstrated that the proposed method was effective for SSL tasks, especially in real situations. In future work, we can apply the proposed method to moving speaker localization and SSL for an unknown number of speakers.

7. Acknowledgement

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2021-0-00456, Development of Ultra-high Speech Quality Technology for Remote Multi-speaker Conference System)

8. References

[1] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 24, no. 4, pp. 320–327, 1976.

[2] J. H. Dibiase, "A high-accuracy, low-latency technique for talker localization in reverberant environments using microphone arrays," Ph.D. thesis, Brown University, 2000.

[3] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas, Prop.*, vol. 34, no. 3, pp. 276–280, 1986.

[4] X. Xiao, S. Zhao, X. Zhong, D. L. Jones, E. S. Chng, and H. Li, "A learning-based approach to direction of arrival estimation in noisy and reverberant environments," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2015, pp. 2814–2818.

[5] S. Chakrabarty and E. A. P. Habets, "Multi-speaker DOA estimation using deep convolutional networks trained with noise signals," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 1, pp. 8–21, 2019.

[6] W. He, P. Motlicek, and J.-M. Odobez, "Deep neural networks for multiple speaker detection and localization," in *Proc. IEEE Int. Conf. Robot., Autom.*, 2018, pp. 74–79.

[7] T. Nguyen, W.-S. Gan, R. Ranjan, and D. L. Jones, "Robust source counting and DOA estimation using spatial pseudo-spectrum and convolutional neural network," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 2626–2637, 2020.

[8] Y. Fu, M. Ge, Y. H., X. Qian, W. L., G. Zhang, and J. Dang, "Iterative sound source localization for unknown number of sources," in *Proc. Interspeech*, 2022, pp. 896–900.

[9] W. He, P. Motlicek, and J.-M. Odobez, "Neural network adaptation and data augmentation for multi-speaker direction-of-arrival estimation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 1303–1317, 2021.

[10] D. Diaz-Guerra, A. Miguel, and J. R. Beltran, "Robust sound source tracking using SRP-PHAT and 3D convolutional neural networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 300–311, 2021.

[11] B. Yang, H. Liu, and X. Li, "SRP-DNN: Learning direct-path phase difference for multiple moving sound source localization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 721–725.

[12] G. K. Papageorgiou, M. Sellathurai, and Y. C. Eldar, "Deep networks for direction-of-arrival estimation in low SNR," *IEEE Trans. Signal Process.*, vol. 69, pp. 3714–3729, 2021.

[13] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Proc. 18th Int. Conf. Artifi. Intell., Statist.*, vol. 38, 2015, pp. 562–570.

[14] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2015, pp. 1395–1403.

[15] Q. Zhu, B. Du, B. Turkbey, P. L. Choyke, and P. Yan, "Deeply-supervised CNN for prostate segmentation," in *Proc. Int. Joint Conf. Neural Netw.*, 2017, pp. 178–184.

[16] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 41–48.

[17] H. Hammer, S. E. Chazan, J. Goldberger, and S. Gannot, "Dynamically localizing multiple speakers based on the time-frequency domain," *EURASIP J. Audio, Speech, Music Process.*, vol. 2021, no. 1, pp. 1–10, 2021.

[18] S. Adavanne, A. Politis, and T. Virtanen, "Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network," in *Proc. Eur. Signal Process. Conf.*, 2018, pp. 1462–1466.

[19] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learning Representations*, 2015.

[20] J. Choi and J.-H. Chang, "Supervised learning approach for explicit spatial filtering of speech," *IEEE Signal Proc. Lett.*, vol. 29, pp. 1412–1416, 2022.

[21] P.-A. Grumiaux, S. Kitić, L. Girin, and A. Guérin, "A survey of sound source localization with deep learning methods," *J. Acoust. Soc. Amer.*, vol. 152, no. 1, pp. 107–151, 2022.

[22] D. Ying, R. Zhou, J. Li, and Y. Yan, "Window-dominant signal subspace methods for multiple short-term speech source localization," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 4, pp. 731–744, 2017.

[23] H. W. Löllmann, C. Evers, A. Schmidt, H. Mellmann, H. Barfuss, P. A. Naylor, and W. Kellermann, "The LOCATA challenge data corpus for acoustic source localization and tracking," in *Proc. IEEE 10th Sens. Array, Multichannel Signal Process. Workshop*, 2018, pp. 410–414.

[24] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2015, pp. 5206–5210.

[25] C. K. A. Reddy, E. Beyrami, J. Pool, R. Cutler, S. Srinivasan, and J. Gehrke, "A scalable noisy speech dataset and online subjective test framework," in *Proc. Interspeech*, 2019, pp. 1816–1820.

[26] D. Diaz-Guerra, A. Miguel, and J. R. Beltran, "gpuRIR: A Python library for room impulse response simulation with GPU acceleration," *Multimedia Tools Appl.*, vol. 80, no. 4, pp. 5653–5671, 2021.

[27] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *J. Acoust. Soc. Amer.*, vol. 65, no. 4, pp. 943–950, 1979.

[28] J. Wiseman, "Wiseman/py-webrtcvad," nov. 2019. [Online]. Available: https://github.com/wiseman/py-webrtcvad.

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