Improved feature extraction for CRNN-based multiple sound source localization

Pierre-Amaury Grumiaux
Orange Labs
Cesson-Sévigné, France
pierreamaury.grumiaux@orange.com

Srđan Kitić
Orange Labs
Cesson-Sévigné, France
srdan.kitic@orange.com

Laurent Girin
Univ. Grenoble Alpes, GIPSA-lab
Grenoble-INP, CNRS
Grenoble, France
laurent.girin@grenoble-inp.fr

Alexandre Guérin
Orange Labs
Cesson-Sévigné, France
alexandre.guerin@orange.com

***Abstract*—In this work, we propose to extend a state-of-the-art multi-source localization system based on a convolutional recurrent neural network and Ambisonics signals. We significantly improve the performance of the baseline network by changing the layout between convolutional and pooling layers. We propose several configurations with more convolutional layers and smaller pooling sizes in-between, so that less information is lost across the layers, leading to a better feature extraction. In parallel, we test the system's ability to localize up to 3 sources, in which case the improved feature extraction provides the most significant boost in accuracy. We evaluate and compare these improved configurations on synthetic and real-world data. The obtained results show a quite substantial improvement of the multiple sound source localization performance over the baseline network.**

***Index Terms*—sound source localization, convolutional recurrent neural network, ambisonics, reverberation**

I. Introduction

Sound source localization (SSL) is a challenging task whose performance is crucial when used as a front-end in practical applications such as teleconferencing [1], source separation [2], speech enhancement [3], speech recognition [4] or Human-robot interaction [5]. Classical subspace-based methods rely on the eigenvalue decomposition of the multichannel observed signal covariance matrix to extract the source signal(s) spatial information [6], [7]. Another classical algorithm (SRP-PHAT) uses a beamformer to build an acoustic map with concentration of energy appearing in the direction of arrival (DOA) of a source [8]. GCC-PHAT is another popular technique which estimates the time-difference of arrival (TDOA) for each pair of microphones to derive the DOA [9]. All these methods work well with a single source, and some of them can provide multiple source DOAs, but they are known to perform poorly in noisy and reverberant environment.

Recently, machine learning methods have greatly improved the performance of SSL systems. In particular, many methods based on deep neural networks (DNNs) have been proposed for estimating the DOA of one source [10], [11], [13], [21] or multiple sources [12], [14], [17], leading to impressive SSL performance under challenging conditions. DNN-based SSL methods can differ on different aspects, starting with the DNN architecture: some methods propose to use a multi-layer perceptron (MLP) [10], [11], a convolutional neural network (CNN) [12], [13], [16], a convolutional recurrent neural network (CRNN) [14], [17], [21] or an autoencoder (AE) [15]. Different types of input features have also been proposed, such as raw signal waveforms [16], features based on the short-time Fourier transform (STFT) [12], [13], correlation-based features [10], [11], [15], or Ambisonics features [14], [17], [21]. Finally, these works can be split into two categories according to the output type, i.e. classification [11]–[15], [17], [21] or regression [10], [16].

In this work, we propose an extension of the work in [17], which is based on a CRNN and can be considered as a state-of-the-art SSL system in the context of Ambisonics signals. First, we propose to simultaneously estimate the DOA of up to 3 sources, whereas in [17], SSL was limited to 2 sources. Second, we propose and compare several more complex architectures that are able to significantly improve the performance compared to [17]. Although this may look trivial at first sight, it is not, at least from an experimental point of view. In fact, among the many architecture variants we tested, most did *not* lead to improved performance. Thereby, we believe that reporting substantial improvement in SSL performance can be useful to the community and lead to a better understanding of how the spatial information is processed in a CRNN for SSL.

II. Proposed method

In this section, we successively describe the input features, the output configuration and the architecture of the neural networks that we trained and tested in the reported experiments.

A. Input features

As in [17], we work with the Ambisonics signal representation, derived from (true or simulated) recordings on a spherical microphone array [18]. The Ambisonics format is well-suited to represent the spatial properties of a soundfield, and is, to some extent, agnostic to the microphone array configuration [19]. It relies on the decomposition of the soundfield on the orthogonal basis of spherical harmonics. The number of retained coefficient defines the order of the representation: an Ambisonics representation of order m requires a spherical microphone array outputing at least $(m+1)^2$ channels. In our experiments, we used first-order Ambisonics (FOA) ($m = 1$) which has shown to provide sufficient spatial information for single- and multiple-speaker localization based on neural

networks [17], [21]–[24]. The four FOA coefficients are W (order 0 spherical harmonic), which can be seen as an omni-directional microphone at the recording point, and X, Y and Z (order 1 spherical harmonics) which can be seen as three orthogonal bidirectional microphones at the recording point. A plane wave, arriving from a direction given by an azimuth θ and elevation ϕ, is encoded into FOA channels as follows:

$$\begin{bmatrix} W(t,f) \\ X(t,f) \\ Y(t,f) \\ Z(t,f) \end{bmatrix} = \begin{bmatrix} 1 \\ \sqrt{3}\cos\theta\cos\phi \\ \sqrt{3}\sin\theta\cos\phi \\ \sqrt{3}\sin\phi \end{bmatrix} p(t,f), \quad (1)$$

where $p(t,f)$, t and f denote the acoustic pressure, and STFT time and frequency bins, respectively.

In this paper, as in [17], we use the active and reactive intensity vectors, respectively defined by [20]:

$$\boldsymbol{I}_a = \mathrm{Re}\{p(t,f)\boldsymbol{v}^*(t,f)\}, \quad \boldsymbol{I}_r = \mathrm{Im}\{p(t,f)\boldsymbol{v}^*(t,f)\}, \quad (2)$$

where $\boldsymbol{v}$ is the particle velocity. For a plane wave, this latter is given in the FOA representation by [25]:

$$\boldsymbol{v}(t,f) = \frac{1}{\rho_0 c\sqrt{3}} \begin{bmatrix} X(t,f) \\ Y(t,f) \\ Z(t,f) \end{bmatrix}, \quad (3)$$

where ρ_0 is the density of air and c is the speed of sound in the air. The active intensity represents the energy flow in a particular spatial point, while the reactive intensity represents dissipative local energy transfers. Noting that $p(t,f) = W(t,f)$ and disregarding the constant factor, the active and reactive intensity vectors in the FOA representation can be reformulated as (indexes t and f are omitted for concision):

$$\boldsymbol{I}_a = \begin{bmatrix} \mathrm{Re}\{WX^*\} \\ \mathrm{Re}\{WY^*\} \\ \mathrm{Re}\{WZ^*\} \end{bmatrix}, \quad \boldsymbol{I}_r = \begin{bmatrix} \mathrm{Im}\{WX^*\} \\ \mathrm{Im}\{WY^*\} \\ \mathrm{Im}\{WZ^*\} \end{bmatrix}. \quad (4)$$

For each time-frequency (TF) bin, the above STFT-domain active and reactive intensity vectors are concatenated to form a 6-channel vector. This vector is then normalized by dividing it by the sound power given by $|W(t,f)|^2 + \frac{1}{3}(|X(t,f)|^2 + |Y(t,f)|^2 + |Z(t,f)|^2)$, which is reminiscent of the so-called *Frequency Domain Velocity Vector* representation [31]. Then, the normalized vectors are concatenated across time and frequency bins to form a 3D $T \times F \times 6$ input tensor, where T is the number of frames and F is the number of frequency bins. In our experiments, we used signals sampled at 16 kHz, a 1,024-point (64 ms) STFT (hence $F = 513$) with a sinusoidal analysis window and 50% overlap. Each input sequence given to our CRNN contains $T = 25$ frames (i.e. about 800 ms of signal), hence an input tensor is of size $25 \times 513 \times 6$.

B. Output

We choose the classification approach for our experiments, which straightforwardly allows for single or multiple sound source localization. To do that, as in [17], we divide the 2D unit sphere into a quasi-uniform grid. The candidate elevations $\phi_i \in [-90, 90]$ and azimuths $\theta_i^j \in [-180, 180]$ on the grid are given by:

$$\begin{cases} \phi_i = -90 + \frac{i}{I} \times 180 & \text{with } i \in \{0, ..., I\} \\ \theta_i^j = -180 + \frac{j}{J^i+1} \times 360 & \text{with } j \in \{0, ..., J^i\}, \end{cases} \quad (5)$$

where $I = \lfloor \frac{180}{\alpha} \rfloor$ and $J^i = \lfloor \frac{360}{\alpha} cos\phi_i \rfloor$ with α the grid resolution in degrees. Each zone around a point of the spherical grid corresponds to a class. The output target of our neural networks is represented by a vector y of size C with "binary" entries, where C is the total number of classes. If a source is present in a zone corresponding to class c, $y(c) = 1$, otherwise $y(c) = 0$. As we address the multiple source localization problem, y can contain multiple entries set to 1.

C. Network architectures

The architectures we used in the experiments reported in the present paper are extensions/variants of the CRNN proposed in [17], which has 3 convolutional layers, followed by a bidirectional LSTM network (BiLSTM), followed by a fully-connected layer. We aim at improving the SSL performance by modifying different aspects of this architecture. First, we propose to increase the number of convolutional layers to extract more "high-level" features that can be efficiently processed by the BiLSTM. In our experiments, we varied the number of convolutional blocks (each containing 2 successive convolutional layers followed by a max-pooling layer) from 4 to 7. Second, in [17] each convolutional block is followed by a max-pooling operation with a quite large pooling size (1×8 for blocks 1 and 2 and 1×4 for block 3). This was to ensure that the dimension at the output of the last convolutional block is significantly reduced for an efficient processing by the BiLSTM block. However, such a large pooling size can cause an important loss of information through the convolutional layers. In order to alleviate this problem, we propose to exploit our larger number of convolutional layers to i) apply max-pooling only every 2 convolutional layers, and ii) reduce the pooling size (details are given in Section III-B).

Fig. 1 shows the generic architecture of the CRNNs we used in the reported experiments. It is composed of B convolutional blocks, $B \in \{4, 5, 6, 7\}$, followed by two BiLSTM layers, and ends with two fully-connected layers. Each convolutional block b_i ($i \in [1, ..., B]$) is composed of 2 convolutional layers with 64 filters of size 3×3, then a max-pooling layer of size $1 \times P_i$, where P_i values are varied in our experiments. Unlike in [17], where a max-pooling layer is used after each convolutional layers, we use only one max-pooling layer after two convolutional layers to allow the network to extract more high-level features before downsampling. Note that if q_i denotes the second dimension of the output of block b_i, we have: $q_i = \frac{q_{i-1}}{p_i}$.

III. Experiments

A. Data

Our training dataset was generated using synthetic spatial room impulse responses (SRIRs), in the same way as in [17].

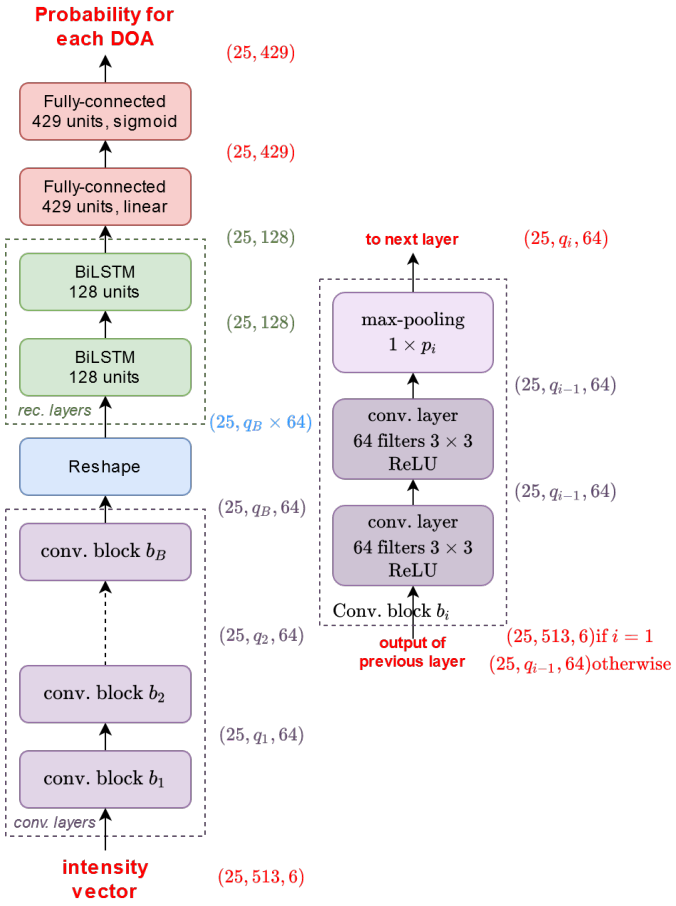

Fig. 1: The proposed improved CRNN architecture. Left: General architecture; Right: detail of one convolutional block. The dimension of the input/output data for each layer is indicated on the right in between layers.

We adapted the SRIR simulator [28] (based on the image-source method [29]), such that it yields the FOA impulse responses. We (randomly) generated many configurations, by varying the size and shape of "shoebox rooms," the reverberation time (RT60), and the source and microphone positions. Microphone signals were obtained by convolving TIMIT speech signals [32] with the simulated SRIRs. For each room configuration, we generated the signals corresponding to a single source, 2- and 3-source mixtures, all 1 s long. Diffuse babble noise was added to those mixtures with a random SNR between 0 and 20 dB. These 1s-long mixtures are finally transformed into 2 sequences of 25 frames with 50% frame overlap. Finally, for our training dataset, we end up with 257,400 sequences for each number of speakers considered in the experiments, that is 1 to 3 speakers, leading to a total of 772,200 training sequences (about 172 hours).

To test our models, we used three types of datasets: i) a dataset based on synthetic SRIRs, that was generated in the same way as for the training but with different SRIRs, speech and noise signals; ii) a dataset based on recorded SRIRs in our acoustic lab (RT60 $\approx$ 500 ms), using all possible combinations

Config.	# parameters	P_1	P_2	P_3	P_4	P_5	P_6	P_7
4-2	700,259	8	4	4	2	-	-	-
4-4	765,795	4	4	4	2	-	-	-
4-8	896,867	4	4	2	2	-	-	-
5-2	774,315	4	4	4	2	2	-	-
5-4	839,851	4	4	2	2	2	-	-
6-2	848,371	4	4	2	2	2	2	-
6-4	913,907	4	2	2	2	2	2	-
7-2	922,427	4	2	2	2	2	2	2
7-4	987,963	2	2	2	2	2	2	2

TABLE I: Max-pooling sizes P_i of the successive convolutional blocks b_i of the improved CRNN. The corresponding total number of parameters is given in column 2.

from 36 microphone positions and 16 loudspeaker positions; iii) the real-world evaluation dataset from the LOCATA challenge [30], with raw Eigenmike recordings converted into the FOA format. We considered LOCATA Task 1 (single static source), Task 2 (multiple static sources), Task 3 (single moving source) and Task 4 (multiple moving sources).

B. Configurations

Table I details the CRNN configurations used in our experiments. As stated before, the configurations differ in the number of convolutional blocks B and the max-pooling size P_i of each convolutional block b_i. The tested combinations of P_i values are such that the size of the data second dimension after the last max-pooling layer q_B is equal to 2, 4 or 8. For concision, a configuration name is of the form "B-q_B", e.g. 4-2 stands for $B = 4$ and $q_B = 2$.

C. Training procedure

The CRNNs were designed with Keras and trained using the Nadam optimizer [27] with default parameters and Nvidia GTX1080 GPUs. Early stopping was applied with a patience of 20 epochs and the learning rate was divided by 2 with a patience of 10 epochs, both by monitoring the accuracy on the validation set. The maximum number of epochs was 300.

D. Evaluation procedure

When inferring the DOAs in test examples, we average the output of the trained network in the frame dimension, i.e. we obtain one probability value per DOA for a 25-frame sequence. Unlike in [17], we do not smooth the probability distribution within a neighborhood since we found out that the results were degraded compared to using the raw distribution. Instead, we directly keep the S highest peaks where S is the number of sources (a peak represents the local maximum of probability distribution within the spherical geometry). In the first two experiments, we suppose that S in known, while for the experiments based on the LOCATA dataset we use the fixed threshold $\beta = 0.2$ to detect source(s).

E. Metrics and baselines

We evaluated the DOA estimation in terms of sequence-wise accuracy, i.e. the percentage of 25-frame sequences whose DOA yields to an angular error less than a certain tolerance. Here, the tolerance we used was either 10° or 15°, considering

Model	1 source				2 sources				3 sources			
	Acc. <10°	Acc. <15°	Mean	Med.	Acc. <10°	Acc. <15°	Mean	Med.	Acc. <10°	Acc. <15°	Mean	Med.
Baseline [17]	94.6	99.2	5.2	4.7	77.6	85.7	15.3	6.0	57.1	68.1	27.2	8.3
4-2	97.6	99.6	4.7	4.2	86.7	92.5	8.3	4.9	71.4	79.9	15.0	6.3
4-4	98.3	**99.7**	4.5	**4.1**	87.9	92.7	8.5	4.8	71.2	79.5	15.4	6.2
4-8	98.2	99.6	4.5	**4.1**	88.0	92.6	8.4	4.8	72.2	80.7	14.7	6.1
5-2	98.3	**99.7**	4.6	**4.1**	87.9	92.8	8.0	4.7	72.5	80.5	14.6	6.1
5-4	98.4	**99.7**	4.6	**4.1**	88.8	93.1	8.1	4.7	73.3	81.0	14.9	6.1
6-2	98.4	99.5	4.7	**4.1**	88.7	93.2	8.1	4.8	72.2	80.7	14.4	6.1
6-4	**98.6**	**99.7**	**4.4**	**4.1**	88.3	93.3	**7.7**	4.7	**74.7**	**83.4**	**12.8**	5.9
7-2	97.8	99.5	4.7	**4.1**	86.3	92.0	8.6	4.8	68.6	77.4	16.3	6.5
7-4	98.4	**99.7**	**4.4**	**4.1**	**89.2**	**93.5**	7.8	**4.6**	74.4	81.3	14.1	**5.8**

TABLE II: SSL results on the test dataset generated with synthetic SRIRs (best results are in bold).

Model	1 source				2 sources				3 sources			
	Acc. <10°	Acc. <15°	Mean	Med.	Acc. <10°	Acc. <15°	Mean	Med.	Acc. <10°	Acc. <15°	Mean	Med.
Baseline [17]	75.2	91.9	8.3	6.3	59.8	75.2	16.7	8.3	44.3	58.4	26.2	11.9
4-2	77.7	92.9	8.1	**6.1**	67.5	83.6	12.9	7.4	53.3	67.5	21.3	9.2
4-4	77.7	93.2	7.9	**6.1**	66.7	83.4	12.9	7.5	54.0	69.2	20.4	9.1
4-8	77.8	92.7	8.1	**6.1**	69.2	84.1	12.5	7.2	54.8	69.7	20.5	9.1
5-2	77.0	93.6	7.9	6.2	68.1	84.1	12.1	7.3	55.3	69.6	18.7	8.9
5-4	78.7	**93.8**	**7.6**	6.2	**70.2**	86.0	12.0	**7.0**	55.4	70.9	19.7	8.9
6-2	78.1	93.6	**7.6**	**6.1**	68.5	84.8	11.9	7.1	53.9	69.3	19.7	9.1
6-4	79.0	93.7	**7.6**	**6.1**	68.2	84.7	11.9	7.2	56.8	**73.3**	**17.3**	8.7
7-2	76.6	93.4	7.7	6.3	68.0	83.7	12.2	7.2	53.3	66.8	20.9	9.3
7-4	**79.8**	93.6	7.7	**6.1**	68.6	**86.2**	**11.7**	7.2	**56.9**	70.7	19.6	**8.6**

TABLE III: SSL results on the dataset generated with real SRIRs (best results are in bold).

Model	Task 1					Task 2					Task 3					Task 4				
	Accuracy			Mean	Median	Accuracy			Mean	Median	Accuracy			Mean	Median	Accuracy			Mean	Median
	<10°	<15°	<20°			<10°	<15°	<20°			<10°	<15°	<20°			<10°	<15°	<20°		
Baseline [17]	35.8	50.7	96.6	13.5	14.9	30.5	68.2	91.8	13.9	14.1	29.5	66.9	86.9	14.1	12.7	30.3	56.7	85.6	15.1	14.1
4-2	40.0	51.7	99.2	13.0	13.0	26.9	71.7	91.9	14.0	12.9	36.0	71.2	88.5	13.4	12.0	55.2	73.1	89.9	12.5	8.6
4-4	40.9	50.6	98.8	13.3	13.0	**34.5**	69.7	91.6	13.4	13.0	40.1	70.0	88.4	12.4	11.8	53.9	71.0	91.2	12.2	8.4
4-8	36.4	51.5	91.0	13.5	14.9	29.1	73.3	91.5	14.0	13.0	41.3	72.3	91.7	12.0	11.5	52.2	70.9	90.8	11.9	9.3
5-2	39.2	49.7	99.0	13.5	16.1	27.0	70.1	93.1	13.6	13.6	40.2	72.5	91.7	11.9	11.7	52.0	70.1	91.6	11.8	9.2
5-4	**46.5**	51.2	98.4	**12.4**	**12.0**	32.8	**73.4**	92.5	13.4	**12.5**	45.3	77.7	91.3	11.8	**10.8**	56.2	73.6	91.7	11.7	8.1
6-2	35.3	52.5	95.8	13.7	14.5	22.4	67.4	90.4	14.6	13.7	35.8	71.7	90.9	13.7	12.1	49.7	68.1	89.2	12.7	10.1
6-4	23.5	49.4	98.3	14.4	15.8	19.1	66.5	92.4	14.5	13.7	28.7	62.6	88.4	13.7	13.3	38.2	59.8	88.3	13.8	13.6
7-2	32.9	**55.5**	91.6	16.0	13.0	23.2	69.3	91.3	14.7	13.6	35.8	71.0	86.9	15.7	12.2	48.8	69.0	89.3	12.8	10.3
7-4	43.6	49.4	**99.4**	12.7	16.1	32.0	71.5	**93.8**	**13.0**	13.0	**45.6**	**75.3**	**92.5**	**11.3**	**10.8**	**58.8**	**74.0**	**92.0**	**11.1**	**7.6**

TABLE IV: SSL results on the LOCATA evaluation dataset (best results are in bold).

the fact that the minimum angle between two points in our grid is 7°. We also evaluated the mean and median angular error, averaged over all test sequences. We compared the proposed improved CRNN (with different configuration settings) and the CRNN of [17] which, again, can be considered as a state-of-the-art baseline (indeed, this baseline network was shown in [17] to perform better than the algorithm proposed in [33], with the latter known for outperforming the LOCATA baseline [30]). Note that in [17], training and test is made for up to 2 speakers. For fair comparison, we trained their network on our dataset with up to 3 speakers.

F. Results

Tables II and III show the performance of all the tested architectures and the baseline CRNN on the dataset generated with synthetic SRIRs and the one generated with real SRIRs. Overall, we observe that the accuracy is better for all architectures compared to the baseline. the new architectures provide substantial improvement in performance over the baseline network, with the best performance obtained by models 6-4 and 7-4: for 1-source, model 6-4 reaches an accuracy (10° tolerance) of 98.6% with a mean angular error of 4.4° on data with synthetic SRIRs and 79.0% with a mean error of 7.6° on for real SRIRs; for 2-source, the accuracy of model 7-4 is 89.2% with a mean error of 7.8° and 68.6% with a mean error of 11.7° for synthetic and recorded SRIRs respectively; for 2-source, 6-4 performs at 74.7% accuracy and 12.8° mean error (synthetic SRIRs) and 56.8% accuracy and 17.3° mean error (real-world SRIRs). For those models, the absolute increase in performance is more than 11% and 8% (synthetic and real SRIRs respectively) for 2-source signals and more than 17% and 12% for 3-source signals, which represents a relative improvement of about 14% on 2-source and 28% on 3-source, which is a substantial gain in performance. The same tendency can be observed looking at the accuracy for an angle tolerance of 15°. As we notice only a slight improvement for 1-source signals, the baseline CRNN seems already well-tuned for monosource localization. However, we can clearly see that the proposed use of more convolutional blocks with less pooling allows a better feature extraction for multi-source localization. The substantial improvement in the mean angular error, with a slight drop of median angular error, means that the proposed architectures allow the network for a more robust multi-source localization, with less outliers (*i.e.* with less estimates being very far from the ground-truth).

The results obtained on the LOCATA dataset are given in Table IV. While we notice a classical drop in performance for all tested models compared to synthetic data, these results confirm the trend observed in the previous two experiments, as the best overall performance is achieved by the 7-4 model. Interestingly, the largest gains in accuracy (with respect to the baseline), as well as in the reduction of mean and median angular errors, are observed in the tasks 3 and 4 which involve mobile sources.

In a general manner, the largest tested model (7-4) performs best in our experiments with respect to several metrics, but the results for smaller architectures are not far behind. For example, Model 6-4 performs better than 7-2. We see a slight tendency that the higher the q_B dimension, the better the results. However it is hard to conclude whether increasing q_B is better for multi-source localization, and a deeper investigation might be necessary in this regard.

IV. Conclusion

We have modified the feature extraction part of a state-of-the-art CRNN-based source localization neural network, leading to substantial performance improvements, as observed in new experiments using synthetic and real-world data. While the improvement in the single-source scenario is incremental, the novel architectures largely outperform the baseline in the more challenging multi-source setting. Future work will focus on structural modifications of the remaining parts of the CRNN architecture, namely by adapting the recurrent, and/or the network output layers.

References

[1] S. Zhao, S. Ahmed, Y. Liang, K. Rupnow, D. Chen, and DL. Jones, “A real-time 3D sound localization system with miniature microphone array for virtual reality,” IEEE ICIEA, pp. 1853–1857, 2012.

[2] S. E. Chazan, H. Hammer, G. Hazan, J. Goldberger, and S. Gannot, "Multi-Microphone Speaker Separation based on Deep DOA Estimation," European Signal Processing Conference (EUSIPCO), 2019.

[3] A. Xenaki, J. B. Boldt, and M. G. Christensen, "Sound source localization and speech enhancement with sparse Bayesian learning beamforming," The Journal of the Acoustical Society of America, vol. 143, no. 6, pp. 3912-3921, 2018.

[4] H. Y. Lee, J. W. Cho, M. Kim, and H. M. Park, "DNN-based feature enhancement using DOA-constrained ICA for robust speech recognition," IEEE Signal Processing Letters, vol. 23, no 8, pp. 1091-1095, 2016.

[5] X. Li, L. Girin, F. Badeig, and R. Horaud, “Reverberant sound localization with a robot head based on direct-path relative transfer function,” IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016.

[6] R. O. Schmidt, “Multiple emitter location and signal parameter estimation,” IEEE Trans. Antennas Propagat., vol. 34, no. 3, pp. 276–280, 1986.

[7] R. Roy, and T. Kailath, "ESPRIT - Estimation of Signal Parameters via Rotational Invariance Techniques," IEEE Trans. Acoust. Speech Signal Process., vol. 37, no. 7, 1989.

[8] J. Dmochowski, J. Benesty, and S. Affes, “A generalized steered response power method for computationally viable source localization,” IEEE/ACM Trans. Audio, Speech, Lang. Process., vol. 15, no. 8, pp. 2510–2526, 2007.

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

[10] F. Vesperini, P. Vecchiotti, E. Principi, S. Squartini and F. Piazza, "A neural network based algorithm for speaker localization in a multi-room environment," IEEE International Workshop on Machine Learning for Signal Processing (MLSP), 2016.

[11] 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," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015.

[12] S. Chakrabarty and E. A. P. Habets, "Multi-speaker DOA estimation using deep convolutional networks trained with noise signals," IEEE Journal of Selected Topics in Signal Processing, vol. 13, no. 1, pp. 8-21, 2019.

[13] N. Yalta, K. Nakadai, and T. Ogata, “Sound source localization using deep learning models,” J. Robot. Mechatron., vol. 29, no.1, pp. 37-48, 2017.

[14] S. Adavanne, A. Politis and T. Virtanen, "Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network," European Signal Processing Conference (EUSIPCO), 2018.

[15] Z. Liu, C. Zhang and P. S. Yu, "Direction-of-arrival estimation based on deep neural networks with robustness to array imperfections," IEEE Transactions on Antennas and Propagation, vol. 66, no. 12, pp. 7315-7327, 2018.

[16] J. Vera-Diaz, D. Pizarro, and J. Macias-Guarasa, "Towards end-to-end acoustic localization using deep learning: From audio signals to source position coordinates," Sensors, vol. 18, no. 10, p. 3418, 2019.

[17] L. Perotin, R. Serizel, E. Vincent, and A. Guérin, “CRNN-based multiple DoA estimation using acoustic intensity features for Ambisonics recordings”, IEEE Journal of Selected Topics in Signal Processing, vol. 13, no. 1, pp. 22–33, 2019.

[18] F. Zotter and M. Frank, "Ambisonics: A practical 3D audio theory for recording, studio production, sound reinforcement, and virtual reality," Springer, 2019.

[19] J. Daniel, “Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia”, Paris VI, 2001.

[20] F. Jacobsen, "A note on instantaneous and time-averaged active and reactive sound intensity," Journal of Sound and Vibration, vol. 147, no. 3, p. 489-496, 1991.

[21] L. Perotin, R. Serizel, E. Vincent, and A. Guérin, “CRNN-based joint azimuth and elevation localization with the Ambisonics intensity vector”, International Workshop on Acoustic Signal Enhancement (IWAENC), 2018.

[22] A. Fahim, P. N. Samarasinghe, and T. D. Abhayapala, “Multi-source DOA estimation through pattern recognition of the modal coherence of a reverberant soundfield,” IEEE/ACM Transactions on Audio, Speech and Language Processing, vol. 28, pp. 605-618, 2019.

[23] S. Adavanne, A. Politis, and T. Virtanen. "Localization, detection and tracking of multiple moving sound sources with a convolutional recurrent neural network", arXiv:1904.12769, 2019.

[24] D. Comminiello, M. Lella, S. Scardapane, and A. Uncini, "Quaternion convolutional neural networks for detection and localization of 3D sound events", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019.

[25] V. Pulkki, S. Delikaris-Manias, and A. Politis, "Parametric time-frequency domain spatial audio," John Wiley & Sons, 2017.

[26] M. Baqué, “Analyse de scène sonore multi-capteurs,” Ph.D. dissertation, Univ. du Maine, 2017.

[27] T. Dozat, “Incorporating Nesterov momentum into Adam,” Univ. of Stanford, Tech. Rep., 2015.

[28] E. A. P. Habets, “Room impulse response generator,” Technische Universiteit Eindhoven, Tech. Rep., 2006.

[29] J. B. Allen and D. A. Berkley, “Image method for efficiently simulating small-room acoustics,” J. Acoust. Soc. Am., vol. 65, no. 4, pp. 943–950, 1979.

[30] C. Evers, H. W. Löllmann, H. Mellmann, A. Schmidt, H. Barfuss, P. A. Naylor, W. Kellermann, “The LOCATA Challenge: Acoustic source localization and tracking,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 1620-1643, 2020.

[31] J. Daniel, and S. Kitić, “Time-domain velocity vector for retracing the multipath propagation,” IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020.

[32] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren, and V. Zue, “TIMIT acoustic–phonetic continuous speech corpus,” Linguistic Data Consortium, 1993.

[33] S. Kitić, and A. Guérin, “TRAMP: Tracking by a real-time AMbisonic-based Particle filter,” IEEE-AASP Challenge on Acoustic Source Localization and Tracking (LOCATA), 2018.