DOI: 10.61782/fa.2023.1132

forum acusticum 2023

PERMUTATION INVARIANT RECURRENT NEURAL NETWORKS FOR SOUND SOURCE TRACKING APPLICATIONS

David Diaz-Guerra[1*] **Archontis Politis**[1] **Antonio Miguel**[2]
Jose R. Beltran[2] **Tuomas Virtanen**[1]
[1] Audio Research Group, Tampere University, Finland
[2] Department of Electronic Engineering and Communications, University of Zaragoza, Spain

ABSTRACT

Many multi-source localization and tracking models based on neural networks use one or several recurrent layers at their final stages to track the movement of the sources. Conventional recurrent neural networks (RNNs), such as the long short-term memories (LSTMs) or the gated recurrent units (GRUs), take a vector as their input and use another vector to store their state. However, this approach results in the information from all the sources being contained in a single ordered vector, which is not optimal for permutation-invariant problems such as multi-source tracking. In this paper, we present a new recurrent architecture that uses unordered sets to represent both its input and its state and that is invariant to the permutations of the input set and equivariant to the permutations of the state set. Hence, the information of every sound source is represented in an individual embedding and the new estimates are assigned to the tracked trajectories regardless of their order.

Keywords: *Sound source tracking (SST), permutation-invariant recurrent neural networks (PI-RNN)*

1. INTRODUCTION

In recent years, the state-of-the-art of sound source localization established by classic signal processing techniques has been surpassed by new systems using deep-learning models [1]. These models use different input features and network architectures, but most of them track the temporal evolution of the signals using convolutional layers followed by recurrent layers [2–4]. Using these architectures, the latent representations at every hidden layer are difficult to interpret and we cannot exploit the permutation invariance of the tracking problem where, if we cannot apply any criteria to order or classify the sources, any permutation of the sources should be considered equally correct.

In [5], we proposed an icosahedral convolutional neural network (icoCNN) for single source localization where the output of the last convolutional layer can be interpreted as the probability distribution of the direction of arrival (DOA) and we can obtain the estimated DOA as its expected value. Extending this model to multi-source scenarios is straightforward and we just need to increase the number of channels of the last convolutional layers to the maximum number of concurrent sources M that the model should be able to localize. Following this approach, after computing the expected value of every one of the M probability distributions generated by the icoCNN, we obtain a set of M DOAs that should be considered invariant to the permutations of its elements. In order to incorporate a recurrent neural network (RNN) after the localization model to increase its temporal perceptive field and improve its tracking capabilities, we could concatenate every element of the DOA set into a single vector and use it as the input of a gated recurrent unit (GRU) [6] or a long short-term memory (LSTM) layer [7]. However, we should expect the output of a tracking system to not be affected by the order of the new estimates at every time frame (i.e., to be invariant to the permutations of the input set), and a conventional RNN operating over the concatenation of the es-

*__Corresponding author__: *david.diaz-guerra@tuni.fi.*

timates would need to learn this property during the training instead of being part of its architecture. In addition, in a tracking system, we can also expect the association of a new estimate to the tracked trajectories be done regardless of their order (i.e., be equivariant to the permutations of the state set) but the state vector generated by a conventional RNN would contain the information of every tracked trajectory in an unstructured way so we would not be able to exploit this property either.

In this paper we present a permutation-invariant recurrent neural network (PI-RNN) that takes an unordered set of embeddings as input (each one with the information of one of the sources detected by the localization network) and generates a recursive output, or state, that is also an unordered set of embeddings with the information of every tracked trajectory. As we could expect from a tracking system, the proposed architecture associates the embeddings in the input set to the embeddings of the state set in a way that is invariant to the permutations of the input set and equivariant to the permutations of the state set.

To the best of our knowledge, this is the first recurrent layer that works with sets instead of with vectors. The closest proposal in the literature is probably the TrackFormer [8], a model for multiple object tracking on video signals that is based on the DETR transformer [9, 10], a model for object detection on images. The recursivity of the TrackFormer model is built around the decoder of the DETR transformer by using the output obtained for a video frame as the input for the following frame. Compared with the TrackFormer, the PI-RNN is not a model but a layer that can be integrated easily into many different models. In addition, it is based on an architecture, the conventional GRU, that, unlike the transformer, was designed to be used in recurrent loops.

Thanks to be taking into account the symmetries of the problem, the proposed PI-RNN, compared with the conventional RNN, scales better with the number of tracked sources and the amount of information stored in each one. Furthermore, we present experiments proving that they can obtain better tracking results than the conventional GRUs.

2. NETWORK ARCHITECTURE

Conventional RNNs use a $\mathbf{h}(t) \in \mathbb{R}^{d_h}$ vector to store the tracking state, which is updated at every time frame based on an input vector $\mathbf{x}(t) \in \mathbb{R}^{d_x}$ using fully connected perceptrons whose computational complexity and number of trainable parameters grow linearly with d_x and quadratically with d_h. When applied to track up to M sources, the information of all the sources and tracked trajectories are stored in these vectors without any structure, so there is a trade-off between the number of sources M that we can track, the amount of information that we store about each one, and the model size and complexity.

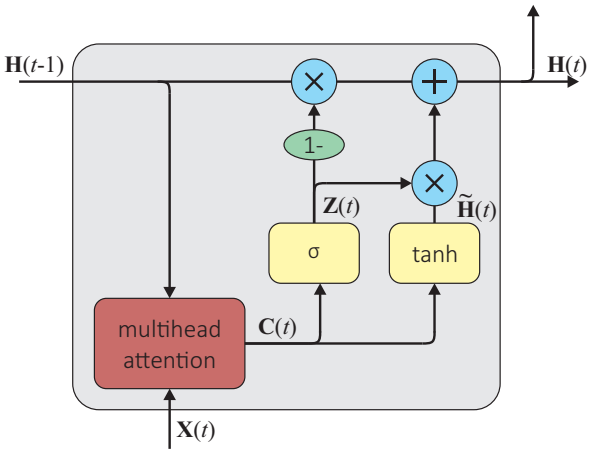

Figure 1. Architecture of the proposed permutation invariant recurrent layer.

In contrast to conventional RNNs, we propose to replace the input and state vectors $\mathbf{x}(t)$ and $\mathbf{h}(t)$ with the sets of embedding $\mathbf{X}(t) = \{\mathbf{x}_1(t), \mathbf{x}_2(t), ..., \mathbf{x}_{M_X}(t)\}$ and $\mathbf{H}(t) = \{\mathbf{h}_1(t), \mathbf{h}_2(t), ..., \mathbf{h}_{M_H}(t)\}$ where every element $\mathbf{x}_i(t) \in \mathbb{R}^{d_x}$ and $\mathbf{h}_i(t) \in \mathbb{R}^{d_h}$ contains information about a single input detection or tracked trajectory respectively. For the sake of simplicity, we will keep $M_X = M_H = M$ and $d_x = d_h = d$ during the rest of the paper, however the proposed architecture can work with $M_X \neq M_H$ or even with dynamic values that change during time, and it can be easily extended to configurations with $d_x \neq d_h$.

In order to match every new embedding of the input set with the embeddings of the state set, we can use a multi-head attention module [11], which is well known for its use in transformer models and is invariant to the permutation of the elements of its input sets:

$$\begin{aligned}\mathbf{C}(t) = \text{MultiHead}(&\mathbf{H}(t-1),\\ &\mathbf{X}(t) \cup \mathbf{H}(t-1),\\ &\mathbf{X}(t) \cup \mathbf{H}(t-1))\end{aligned} \quad (1)$$

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\mathbf{head}_1, ..., \mathbf{head}_{N_{\text{heads}}}) \quad (2)$$

$$\mathbf{head}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) \quad (3)$$

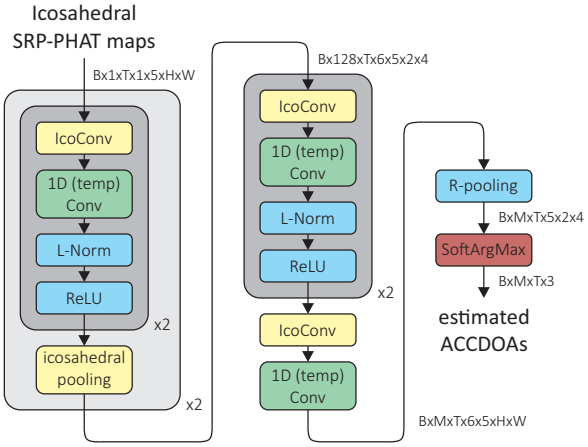

Figure 2. Architecture of the icoCNN used for evaluation. B is the batch size, T is the number of temporal frames of the acoustic scenes, $H = 2^r = 8$ and $W = 2^{r+1} = 16$ are the height and the width of the projections of the icosahedral grid.

$$\text{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i) = \text{soft-max}\left(\frac{\mathbf{Q}_i \mathbf{K}_i^T}{\sqrt{d_k}}\right)\mathbf{V}_i, \quad (4)$$

with the soft-max$(\cdot)$ operating across rows. With this configuration, the generated set $\mathbf{C}(t)$ is invariant to the permutations of $\mathbf{X}(t)$ and equivariant to the permutations of $\mathbf{H}(t-1)$ as we would expect from a tracking system.

Finally, as shown in Fig. 1, once we have assigned the input embeddings to their corresponding state embedding, we can just update every element of the state set according to this assignation:

$$\mathbf{h}_i(t) = [1 - \mathbf{z}_i(t)] \odot \mathbf{h}_i(t-1) + \tilde{\mathbf{h}}_i(t) \quad (5)$$

$$\mathbf{z}_i(t) = \sigma(\mathbf{c}_i(t)\mathbf{W}^z) \quad (6)$$

$$\tilde{\mathbf{h}}_i(t) = \tanh(\mathbf{c}_i(t)\mathbf{W}^h), \quad (7)$$

where $\odot$ denotes element-wise vector multiplication and $\sigma(\cdot), \tanh(\cdot)$ denote sigmoid and hyperbolic tangent functions respectively, applied to each element of their vector arguments. This gated architecture is based on a simplified version of the minimal gated recurrent unit [12], but we could design different architectures based on different conventional recurrent architectures. As in conventional RNNs, the number of trainable parameters grows quadratically with d, but in the case of the PI-GRUs we have M embeddings of size d containing the information of every tracked trajectory. Hence, we can expect our model to scale better when we increase the number of sources we want to track or the amount of information that we want to be able to represent for each one of them.

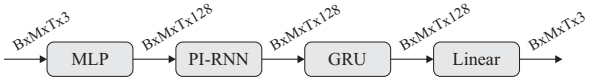

Figure 3. Architecture of the PI-RNN used after the icoCNN in the evaluated model.

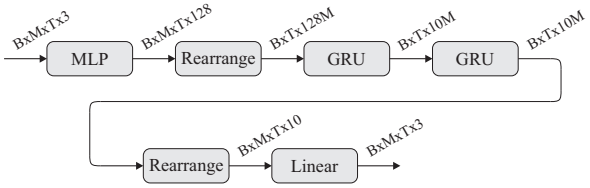

Figure 4. Architecture of the conventional RNN used after the icoCNN in the baseline model.

3. EVALUATION

3.1 Experiment design

As a preliminary study of the performance of this new architecture, we decided to add a PI-RNN after the icoCNN presented in [5] for single source localization. As shown in Fig. 2, in order to extend the icoCNN to multi-source localization, we just increased the number of output channels from 1 to M. Fig. 3 represents the PI-RNN we used after the icoCNN: we first used a multi-layer perceptron to project every ACCDOA [13] generated by the icoCNN into an embedding of size d and then we used those embeddings as the input set of our PI-RNN. After the PI-RNN had associated every new estimate from the icoCNN to one of the tracked trajectories, we added a conventional GRU (operating independently over the embedding of every tracked trajectory so it did not break the permutation invariance of the model) and, finally, we used a linear layer to project the d-size embedding into a 3D ACCDOA. The initial state of every embedding of the state set of the PI-RNN was learned during the training of the model while, at every time frame, the embeddings of all the inactive trajectories were reset (i.e., those who had lead to ACCDOAs with a norm lower than 0.5).

The method was compared to two baselines, a) the icoCNN without any kind of recurrent layers, and b) the icoCNN with two conventional GRUs designed to have

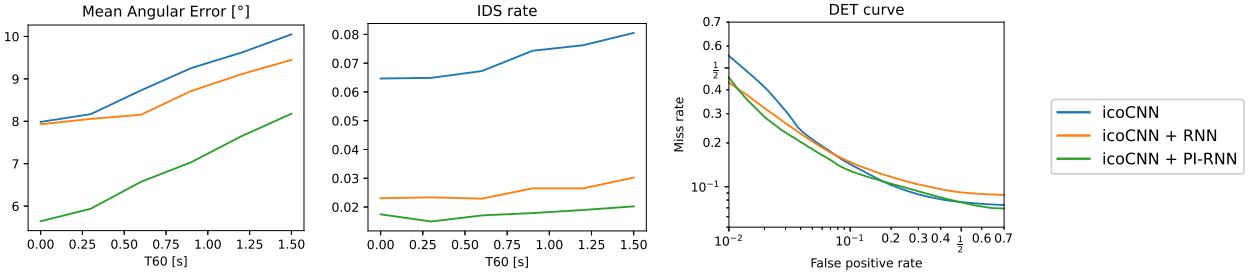

Figure 5. Evaluation metrics for proposed PI-RNN and the baseline models.

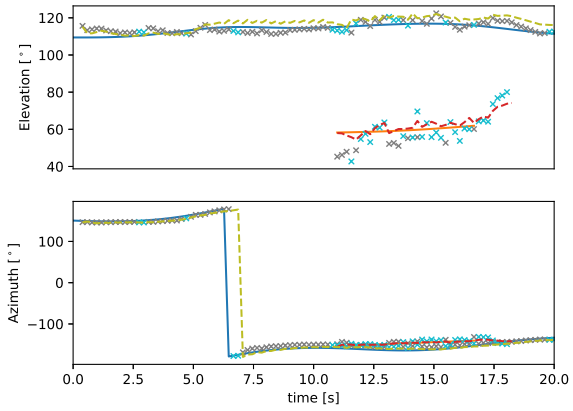

Figure 6. Example of one of the test acoustic scenes. The solid line represents the ground truth trajectories of the sources, the crosses the defections estimated at the icoCNN (i.e., the input of the PI-RNN), and the dashed line the trajectories estimated by the whole model (i.e., the output of the PI-RNN). The color indicates to which of the outputs they correspond, so the IDS are visible.

a similar number of trainable parameters as the evaluated model (see Fig. 4). In order to avoid identity switches (IDSs) in the tracked trajectories, we trained all the models using sliding permutation invariant training (sPIT) [14]. To facilitate the training of the icoCNN, we added an auxiliary frame-level permutation invariant training (fPIT) at its output in the models that included recurrent layers after it.

We used the same synthetic dataset as in [14], where acoustic sources randomly appeared and disappeared along 20-second-length scenes. As source signals, we used speech utterances from the LibriSpeech corpus and we simulated them following random trajectories in rooms with reverberation times from $T_{60} = 0.2$ to $1.3\,\mathrm{s}$ with the image source method. The maximum number of concurrent active sources in a time frame was 3.

We used $M = 10$ as the number of ACCDOA outputs of all our models since we observed that it was beneficial to use a higher number than the maximum possible number of active sources in the dataset (i.e., 3) and we used $d = 128$ as embedding size for the input and state sets of the PI-RNN. This is a preliminary study of this new architecture and further experiments should be conducted for a better optimization of these hyperparameters.

3.2 Results

As we can see in Fig. 5, the proposed PI-RNN clearly outperforms the baselines in terms of localization error and the frequency of the identity switches while, as we can see in the detection error tradeoff (DET) curve, the tradeoff between false positives and misses remains is for all the evaluated models. It is worth saying that both the conventional and the permutation-invariant RNNs are receiving only spatial information about the estimated sources. By modifying the model to include spectral information in their input we could expect both models to improve their performance, with the PI-RNNs scaling better to the amount of spectral information of each source and therefore being able to better exploit it.

As an example, in Fig. 6 we can see one of the test acoustic scenes. We can see how the output of the icoCNN had a high number of identity switches even when only one source was active but the PI-RNN was able to fix these switches and also reduce the localization error.

3.3 Attention matrices

We can interpret the attention matrix of the multi-head attention module of the PI-RNN as an assignment matrix where each row indicates which elements of the input and state set were employed to compute each element of the output set.

The attention matrix shown in Fig. 7a corresponds to the first frame where a source appeared and we can see how it was detected at the 8th output of the icoCNN (i.e., the 8th input of the PI-RNN) and the PI-RNN assigned it to its 9th output. In the attention matrix of the next time frame (Fig. 7b) we can see that the 9th output of the PI-RNN was computed combining the information of the new estimate at that frame with the corresponding recurrent state. A new source was detected by the icoCNN at its 4th output in the time frame corresponding to Fig. 7c and it was assigned to the 10th output of the PI-RNN. Finally, in Fig. 7d we can see how, after an identity switch at the output of the icoCNN, the PI-RNN was able to assign every new estimate to the correct tracked trajectory fixing the identity switch.

4. CONCLUSIONS

We have presented a new RNN architecture whose input and state are presented with sets instead of vectors and that is invariant to the permutation of the elements of the input and equivariant to the permutations of the elements of the state set. This new architecture is able to exploit the permutation symmetries of the tracking problem and to outperform the conventional RNN in the preliminary experiments presented in this paper. We expect the difference between the performance of the PI-RNNs and the conventional RNNs to become even greater when including more information of every source at their input.

5. REFERENCES

[1] P.-A. Grumiaux, S. Kitić, L. Girin, and A. Guérin, "A survey of sound source localization with deep learning methods," *J. Ac. Soc. America*, vol. 152, pp. 107–151, 2022.

[2] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound event localization & detection of overlapping sources using convolutional recurrent neural networks," *IEEE J. Sel. Topics Sig. Proc.*, vol. 13, pp. 34–48, 2019.

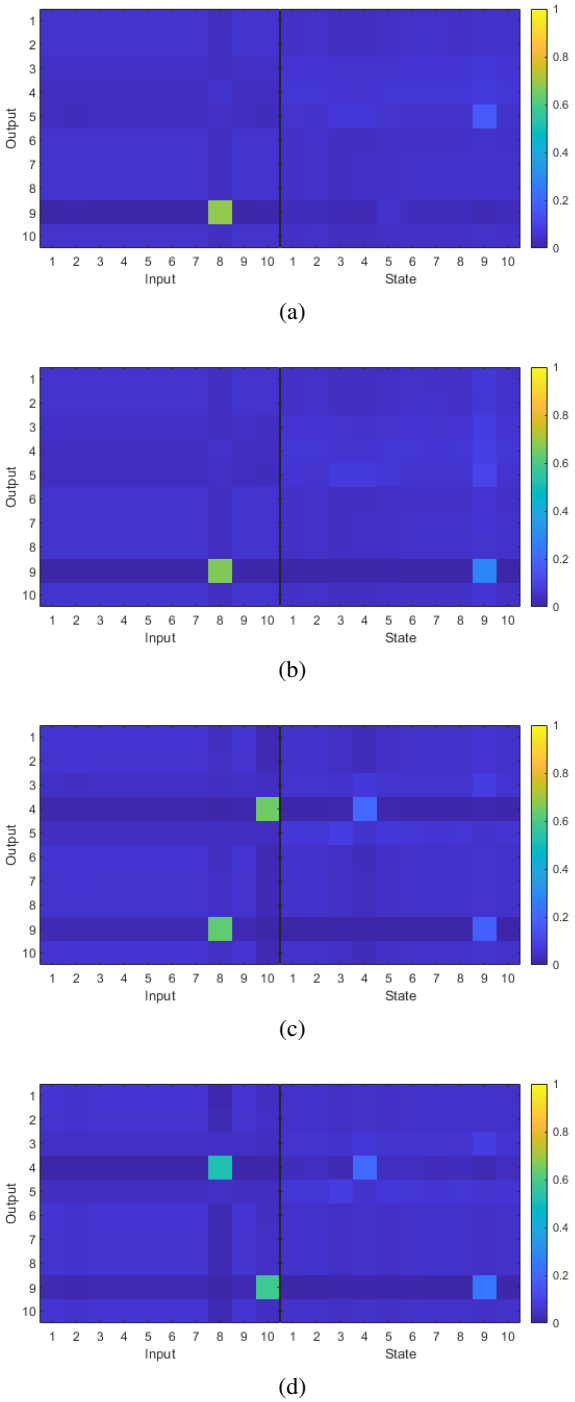

Figure 7. Examples of attention matrices from the multi-head attention module of the PI-RNN.

[3] L. Perotin, R. Serizel, E. Vincent, and A. Guérin, "CRNN-Based Multiple DoA Estimation Using Acoustic Intensity Features for Ambisonics Recordings," *IEEE J. Sel. Topics Sig. Proc.*, vol. 13, pp. 22–33, 2019.

[4] Y. Cao, T. Iqbal, Q. Kong, M. B. Galindo, W. Wang, and M. D. Plumbley, "Two-Stage Sound Event Localization and Detection Using Intensity Vector and Generalized Cross-Correlation," in *Proc. of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, New York University, 2019.

[5] D. Diaz-Guerra, A. Miguel, and J. R. Beltran, "Direction of Arrival Estimation of Sound Sources Using Icosahedral CNNs," *IEEE/ACM Trans. Audio, Speech, and Lang. Proc.*, vol. 31, pp. 313–321, 2023.

[6] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1724–1734, Association for Computational Linguistics, Oct. 2014.

[7] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, pp. 1735–1780, Nov. 1997.

[8] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "TrackFormer: Multi-Object Tracking with Transformers," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (New Orleans, LA, USA), pp. 8834–8844, IEEE, June 2022.

[9] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers," in *Computer Vision – ECCV 2020* (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), Lecture Notes in Computer Science, (Cham), pp. 213–229, Springer International Publishing, 2020.

[10] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable Transformers for End-to-End Object Detection," in *International Conference on Learning Representations*, Feb. 2022.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.

[12] G.-B. Zhou, J. Wu, C.-L. Zhang, and Z.-H. Zhou, "Minimal gated unit for recurrent neural networks," *International Journal of Automation and Computing*, vol. 13, pp. 226–234, June 2016.

[13] K. Shimada, Y. Koyama, N. Takahashi, S. Takahashi, and Y. Mitsufuji, "ACCDOA: Activity-coupled cartesian direction of arrival representation for sound event localization and detection," in *IEEE Int. Conf. on Acoustics, Speech and Sig. Proc. (ICASSP)*, pp. 915–919, 2021.

[14] D. Diaz-Guerra, A. Politis, and T. Virtanen, "Position tracking of a varying number of sound sources with sliding permutation invariant training," in *2023 31th European Signal Processing Conference (EUSIPCO)*, Sept. 2023.