

# EVENT-INDEPENDENT NETWORK FOR POLYPHONIC SOUND EVENT LOCALIZATION AND DETECTION

Yin Cao,<sup>1</sup> Turab Iqbal,<sup>1</sup> Qiuqiang Kong,<sup>2</sup> Yue Zhong,<sup>1</sup> Wenwu Wang,<sup>1</sup> Mark D. Plumbley<sup>1</sup>

<sup>1</sup>Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, UK  
 {yin.cao, t.iqbal, y.zhong, w.wang, m.plumbley}@surrey.ac.uk

<sup>2</sup>ByteDance Shanghai, China  
 kongqiuqiang@bytedance.com

## ABSTRACT

Polyphonic sound event localization and detection is not only detecting what sound events are happening but localizing corresponding sound sources. This series of tasks was first introduced in DCASE 2019 Task 3. In 2020, the sound event localization and detection task introduces additional challenges in moving sound sources and overlapping-event cases, which include two events of the same type with two different direction-of-arrival (DoA) angles. In this paper, a novel event-independent network for polyphonic sound event localization and detection is proposed. Unlike the two-stage method we proposed in DCASE 2019 Task 3, this new network is fully end-to-end. Inputs to the network are first-order Ambisonics (FOA) time-domain signals, which are then fed into a 1-D convolutional layer to extract acoustic features. The network is then split into two parallel branches. The first branch is for sound event detection (SED), and the second branch is for DoA estimation. There are three types of predictions from the network, SED predictions, DoA predictions, and event activity detection (EAD) predictions that are used to combine the SED and DoA features for on-set and off-set estimation. All of these predictions have the format of two tracks indicating that there are at most two overlapping events. Within each track, there could be at most one event happening. This architecture introduces a problem of track permutation. To address this problem, a frame-level permutation invariant training method is used. Experimental results show that the proposed method can detect polyphonic sound events and their corresponding DoAs. Its performance on the Task 3 dataset is greatly increased as compared with that of the baseline method.

**Index Terms**— Sound event localization and detection, direction of arrival, event-independent, permutation invariant training.

## 1. INTRODUCTION

Sound event localization and detection (SELD) has become a more and more popular research topic since DCASE 2019. It detects types of sound events and localizes corresponding sound sources. This year, DCASE 2020 Task 3 [1–4] introduces additional challenges in moving sources and polyphonic cases that include the same class of event but with different direction-of-arrival (DoAs).

For DCASE 2019 Task 3, we introduced a two-stage method for polyphonic SELD [5]. Although we obtained a good ranking, the method was not designed as an actual polyphonic localization method for the reason that it lacks the ability to detect sound events of the same type but with different DoAs. Besides, it is not an elegant end-to-end system, which needs to be trained for two steps.

In this paper, we propose a redesigned event-independent end-to-end system for polyphonic SELD. It is designed for overlapping-event cases, including the presence of the same type of event with different DoAs. It is also convenient to expand the system to the case of more than two overlapping events. The source code is released on GitHub<sup>1</sup>. Our contributions are three-fold. 1) The proposed system predicts overlapping events using track-wise outputs, that is, it predicts event and corresponding DoA for each track. Within each track there may only be maximally one event and corresponding DoA existing. 2) Frame-level permutation invariant training is adopted to solve the track permutation problem. 3) An event activity detection (EAD) prediction is added to combine sound event detection (SED) and DoA estimation feature embeddings to predict the on-set and off-set times more accurately.

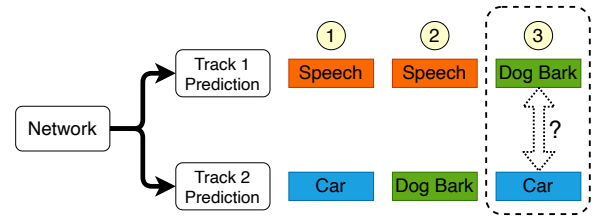


Figure 1: Illustration of the track permutation problem. Numbers mean different group of labels.

We adopt the track-wise prediction to tackle the overlapping-event scenarios. It is needed to pre-determine the number of tracks according to the maximum number of overlapping events. These tracks are event-independent, which means the prediction on each track can be of any type of event. They can even be the same type of event which indicates that two same-type events with different DoAs are predicted. It is also reasonable to assume these tracks are event-independent. Consider a polyphonic prediction case illustrated in Fig. 1. The network has one prediction for each track, within which there could only be maximally one event and corresponding DoA. There are three groups of labels which are all potentially two-event overlapping cases. It is assumed that, for the first group, the “speech” label and the “car” label are tied to track 1 and 2. For the second group, it is reasonable to still assign the “speech” label to track 1, and the new “dog bark” label to track 2. However, for the third group of labels, it is hard to decide to which track the “dog bark” or the “car” label should be assigned. In other words, track permutation problems emerge if track-wise predictions are used.

<sup>1</sup><https://github.com/yinkalarario/EIN-SELD>

To address the track permutation problem, frame-level permutation invariant training (denoted as tPIT) is used. The frame-level permutation invariant training was first proposed for speaker-independent source separation [6, 7]. For our SED problem, the tPIT is implemented by examining all possible track permutations in each frame during training, and then the lowest frame-level loss is selected among these track permutations for the backward propagation to train the model. In this way, the optimal local assignment of track-event pairs can be reached, thus leading to the excellent SED and DoA prediction performance frame-wise.

In order to estimate the frame-level information more accurately, the features from both SED and DoA branches are combined to predict event activities. The aim of this EAD is to constrain the detection of the existence of events (or DoAs) not only from both SED and DoA features. That means SED and DoA predictions have a mutual dependence rather than a one-way dependence. With the proposed system, experimental results show that the performance is greatly increased compared with that of the baseline system.

The rest of the paper is arranged as follows. In Section 3, the proposed learning method is described in detail, including features, network architecture and permutation invariant training. Experimental results and discussions are shown in Section 4. Finally, conclusions are summarized in Section 5.

## 2. RELATED WORKS

### 2.1. Sound Event Localization and Detection

Sound event localization and detection is a novel topic that has wide applications [8]. In DCASE 2019 Task 3 [2, 9], the TAU Spatial Sound Events 2019 dataset was released [10]. There were several innovative papers based on that dataset. Mazzon et al. proposed a spatial-augmentation method which rotates and reflects sources so that unseen DoA data and corresponding labels are produced [11]. Grondin et al. used a CRNN on pairs of microphones to localize and detect sound events [12]. Sundar et al. proposed an encoding scheme to represent spatial coordinates of multiple sources [13]. Although the two-stage method we proposed [5] last year achieved the second-best performance in DCASE 2019 Task 3, it is not an elegant end-to-end system and is not designed for overlapping events of the same type with different DoAs. Then, an idea of similar track-wise prediction was proposed by Nguyen et al. [14]. However, their system did not show a reasonable bond between SED and DoA predictions, they assume the track prediction with the highest probability of SED corresponds to the same highest probability track of DoA. In this paper, it will be shown in the following part that the system proposed is a more complete system in this sense.

### 2.2. Permutation Invariant Training

Permutation invariant training (PIT) was first proposed to tackle the problem of speaker-independent multi-talker speech separation [6], commonly known as the cocktail-party problem. PIT combines the label assignment and minimization together, and can be implemented inside the network structure. It first assigns the best predict-target pairs according to which way the total loss is the smallest, and then minimizes the loss given the assignment. PIT was then extended to frame-level and utterance-level PIT [7, 15–17], which were utilized by a range of speaker-independent speech separation researchers [18–22]. In our proposed method, a track-wise output format is used. Tracks are event-independent, that is, tracks can predict any type

of event. This generates a problem of track permutation. It will be shown that by adopting a similar idea to the frame-level PIT, the track permutation problem can be excellently solved.

## 3. THE METHOD

The proposed method is described in this section. Features used are logmel and intensity vector. They are calculated inside the network using a 1-D convolutional layer. The network architecture and permutation invariant training will be introduced in detail.

### 3.1. Features

In this paper, a logmel spectrogram feature is used for SED, while an intensity vector from first-order ambisonics (FOA) in logmel space is used for DoA estimation. These features are directly calculated using a 1-D convolutional layer instead of being pre-calculated off-line.

FOA, which is also known as B-format, includes four channels of signals,  $w, x, y$  and  $z$ . These four channel signals indicate omni-directional,  $x$ -directional,  $y$ -directional and  $z$ -directional components, respectively. The instantaneous sound intensity vector can be expressed as  $\mathbf{I} = p\mathbf{v}$ , where  $p$  is the sound pressure and can be obtained with  $w$ ,  $\mathbf{v} = (\mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z)^T$  is the particle velocity vector and can be estimated using  $x, y$  and  $z$ . An intensity vector carries the information of the acoustical energy direction of a sound wave, its inverse direction can be interpreted as the DoA, hence the FOA-based intensity vector can be directly utilized for DoA estimation [23–26].

In order to make the intensity vector have the same size as the logmel, it is calculated in the STFT domain and the mel space as

$$\mathbf{I}(f, t) = \frac{1}{\rho_0 c} \Re \left\{ \mathbf{W}^*(f, t) \cdot \begin{bmatrix} X(f, t) \\ Y(f, t) \\ Z(f, t) \end{bmatrix} \right\}, \quad (1)$$

$$\mathbf{I}_{mel}^{\text{norm}}(k, t) = -\mathbf{H}_{mel}(k, f) \frac{\mathbf{I}(f, t)}{\|\mathbf{I}(f, t)\|_2},$$

where,  $\rho_0$  and  $c$  are the density and velocity of the sound,  $W, X, Y, Z$  are the STFT of  $w, x, y, z$ , respectively,  $\Re \{\cdot\}$  indicates the real part,  $*$  denotes the conjugate,  $\|\cdot\|_2$  is a vector's  $\ell_2$  norm,  $k$  is the index of the mel bins, and  $\mathbf{H}_{mel}$  is the mel-band filter banks. In this paper, the three components of the intensity vector are taken as three additional input channels for the network.

### 3.2. Network architecture

The proposed event-independent network uses track-wise outputs. For track-wise predictions, it is needed to pre-determine the number of tracks according to the maximum number of overlapping events. These tracks are event-independent, which means the prediction on each track can be of any type of event.

The network has two branches of feature embeddings, the SED branch and the DoA branch. Its architecture is shown in Fig. 2. FOA time-domain signals are used as inputs and are first fed into two branches. In both of the SED and the DoA branches, a 1-D convolutional layer is first used to extract logmel spectrograms and intensity vectors. They are then normalized by batch-normalization layers. For the SED feature embedding, four groups of convolutional blocks are used to extract the SED embedding. Each convolutional block contains two 2-D convolutional layers with a kernel size of 3x3, a batch-normalization layer, and an average-pooling layer. For the DoA feature embedding, a revised ResNet 18 with two 3x3 2-D convolutional layers as the stem-layer is used. The size of the feature

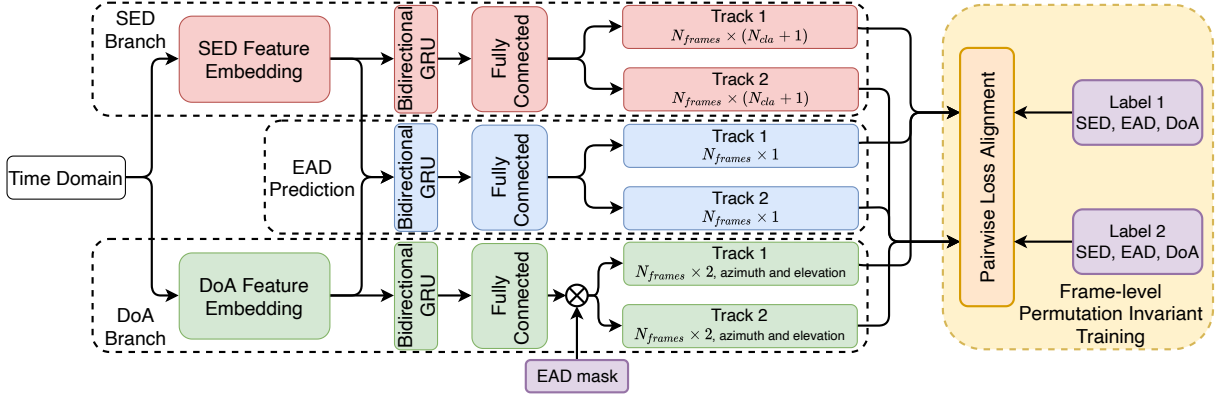


Figure 2: Network Architecture.  $N_{frames}$  is the number of frames.  $N_{cla}$  is the number of classes of events. In the SED branch, there is one additional class of event that is silence. EAD mask block is described in Eq. 4.

maps is 512 in the last layer for both SED and DoA embeddings. These two branches are then used to generate three predictions: SED predictions, EAD predictions, and DoA predictions. For SED and DoA predictions, the respective feature embeddings are fed into a two-layer bidirectional GRU and a fully-connected layer. For EAD predictions, SED and DoA embeddings are combined and are fed into a similar GRU and fully-connected layer. Outputs of the network have a track-wise format. Each track has at most one SED, one EAD, and one DoA prediction.

For SED, predictions have  $N_{cla} + 1$  types of events. Here,  $N_{cla}$  is the number of event classes. The additional class of event indicates the silent class (no event is happening). The softmax activation function is used after each track for SED. The corresponding loss for SED predictions can be written as

$$\ell_t^{\text{SED}}(n_{\text{track}}) = -\log \left[ \frac{\exp(y_t^{\text{SED}}[n_{\text{track}}, \text{class}_t])}{\sum_{j \in \mathbf{J}} \exp(y_t^{\text{SED}}[n_{\text{track}}, j])} \right], \quad (2)$$

$$\mathcal{L}^{\text{SED}} = \sum_{n \in N_{\text{track}}, t} \ell_t^{\text{SED}}(n),$$

where  $\ell_t^{\text{SED}}$  is the track-wise loss.  $\mathcal{L}^{\text{SED}}$  is the total SED loss for updating the model.  $y_t^{\text{SED}}$  denotes the output logits of the SED fully-connected layer,  $t$  indicates the frame,  $n_{\text{track}}$  is the track index,  $\text{class}_t$  is the ground truth target at frame  $t$ ,  $N_{\text{track}}$  is the number of tracks, and  $\mathbf{J}$  is the class set. It is a multi-class single-label problem for each track and a multi-class multi-label problem for all of the tracks.

EAD predictions need to combine SED and DoA embeddings. Binary cross entropy is used as the loss. The existence of EAD predictions is important for three reasons. First, it uses SED and DoA feature embeddings together to predict on-set and off-set information. In this way, the frame-level information does not solely depend on a single branch, label information from both branches can be utilized. In the mean time, the EAD loss can be back propagated to affect both branches of the feature embeddings; second, it constrains SED and DoA feature embeddings to unify track-binding so that tracks do not permute in different branches. That is, track 1 in the SED prediction can always be tied to track 1 in the EAD and DoA predictions. Third, EAD predictions contribute to mask out DoA predictions.

For DoA, predictions for tracks contain azimuth and elevation angles. A linear activation function is used. Under these circumstances, when there is no event happening, the ground truth DoAs

should not be zero. It is therefore reasonable to use a mask to shield those invalid frames. During training, ground truth EAD labels are used as the mask to filter out those frames with actual events happening. Whereas during test, an intersection set of the SED and EAD masks is used. The loss for DoA predictions can be written as Eq. 3,

$$\ell_t^{\text{DoA}}(n_{\text{track}}) = \frac{1}{2} \sum_{\text{azi, elev}} \{ \|y_t^{\text{DoA}} - \hat{y}_t^{\text{DoA}}\|_p \cdot \mathcal{M}_t^{\text{EAD}}(n_{\text{track}}) \}, \quad (3)$$

$$\mathcal{L}^{\text{DoA}} = \frac{1}{\sum_{n \in N_{\text{track}}, t} \mathcal{M}_t^{\text{EAD}}(n)} \sum_{n \in N_{\text{track}}, t} \ell_t^{\text{DoA}}(n),$$

where

$$\mathcal{M}_t^{\text{EAD}} = \begin{cases} \hat{y}_t^{\text{EAD}} & \text{for training,} \\ \mathbb{1}[y_t^{\text{EAD}} > \tau^{\text{EAD}}] \cap \mathbb{1}[y_t^{\text{SED}} = \max_j y_t^{\text{SED}}][0 : N_{cla}] & \text{for test,} \end{cases} \quad (4)$$

is the mask for DoA predictions.  $\ell_t^{\text{DoA}}$  and  $\mathcal{L}^{\text{DoA}}$  are defined the same as the SED loss.  $\hat{y}_t^{\text{DoA}}$  is the DoA ground truth.  $\|\cdot\|_p$  is the  $p$ -norm.  $\hat{y}_t^{\text{EAD}}$  is the EAD ground truth.  $\mathbb{1}[\mathcal{C}]$  is the binarize function, where  $\mathbb{1}[\mathcal{C}]$  is 1 when  $\mathcal{C}$  is True, and 0 when  $\mathcal{C}$  is False.  $\tau^{\text{EAD}}$  is the threshold for EAD.  $\mathbf{J}$  is the class set.  $[0 : N_{cla}]$  means to take the first  $N_{cla}$  items.

### 3.3. Permutation Invariant Training

After track-wise predictions are obtained, frame-level permutation invariant training (tPIT) is adopted to tackle the track permutation problem. In Fig. 2, the tPIT contains the Pairwise-Loss-Alignment block, which assigns labels to different tracks to constitute all of the possible combinations of prediction-label pairs. Then the total loss for each combination is calculated. The lowest total loss is selected as the actual loss to perform the back-propagation. Assume all of the possible combinations of prediction-label pairs constitute a permutation set  $\mathbf{P}$ .  $\alpha(t) \in \mathbf{P}$  is one of the possible permutation pairs at frame  $t$ . The tPIT loss can be written as Eq. 5,

$$\mathcal{L}_t^{\text{tPIT}} = \min_{\alpha(t) \in \mathbf{P}} \sum_{N_{\text{track}}} \{ \ell_{t, \alpha(t)}^{\text{SED}} + \ell_{t, \alpha(t)}^{\text{EAD}} + \ell_{t, \alpha(t)}^{\text{DoA}} \}, \quad (5)$$

where  $\ell_{t,\alpha(t)}^{\text{SED}}$  and  $\ell_{t,\alpha(t)}^{\text{DoA}}$  are defined in Eq. 2 and 3, respectively.  $\ell_{t,\alpha(t)}^{\text{EAD}}$  is the binary cross entropy loss.

Therefore, the process of tPIT is not only to perform the classification or regression training but also to pair the most probable predictions and labels inside the network. In this way, the event-independent track permutation problem can be elegantly solved.

## 4. EXPERIMENTS

In this section, the experimental results using the proposed method on the TAU-NIGENS Spatial Sound Events 2020 dataset are described.

### 4.1. Experimental Setup

The dataset contains 700 sound event samples spread over 14 classes. Within the dataset, realistic spatialization and reverberation through RIRs were collected in 15 different enclosures. From about 1500 to 3500 possible RIR positions across different rooms were applied. The dataset contains both static reverberant and moving reverberant sound events. More information can be obtained in [1, 4].

The evaluation metrics used consider the joint nature of localization and detection [3]. There are two metrics for SED, F-score ( $F_{\leq T^\circ}$ ) and Error Rate ( $ER_{\leq T^\circ}$ ). They consider true positives predicted under a distance threshold  $T = 20^\circ$  from the reference. For the localization part, there are other two metrics which are classification-dependent. The first is a localization error  $LE_{\text{CD}}$  expressing average angular distance between predictions and references of the same class. The second is a localization recall metric  $LR_{\text{CD}}$  expressing the true positive rate of how many of these localization estimates were detected in a class, out of the total class instances.

To generate the weights of the 1-D convolutional layer for feature extraction. A 1024-point Hanning window with a hop size of 480 points is used. Audio clips are segmented to have a fixed length of 5 seconds with 80 % overlap for training. The learning rate is set to 0.0005 for the first 60 epochs and is adjusted to 0.0001 for each epoch that follows. The final results are calculated after 80 epochs. A threshold of 0.5 is used to binarize EAD predictions.

### 4.2. Comparison Systems and Results

In order to assess the performance of the proposed system, an ablation study is performed. Several systems are compared, including:

- **Baseline-FOA:** The baseline method using Ambisonic.
- **Baseline-Mics:** The baseline method using microphone array.
- **Track-Wise 1:** Track-wise output without EAD and tPIT.
- **Track-Wise 2:** Track-wise output with EAD but without tPIT. SED predictions are used as the mask to filter out active tracks.
- **Track-Wise 3:** Track-wise output with EAD but without tPIT. SED and EAD predictions are together used as the mask to filter out active tracks according to Eq. 4.
- **Event-Ind:** The proposed event-independent system with EAD and tPIT. SED and EAD predictions are together used as the mask.

All of the experimental results were evaluated on fold 1 and were averaged with 5 different trials. Results for comparison are shown in Fig. 3. It can be seen that all of the proposed methods are better than the baselines with two-stage method [5]. A simple comparison for ablation study shows that without EAD and tPIT, the “Track-Wise 1” method is the worst among proposed methods. There seems to be an exception of  $LE_{\text{CD}}$ , where “Track-Wise 1” method is even lower

than “Track-Wise 2” and “Track-Wise 3”. This is due to the reason that during experiments, it was found that there is a trade-off between  $LR_{\text{CD}}$  and  $LE_{\text{CD}}$ . That is, when  $LR_{\text{CD}}$  is increased,  $LE_{\text{CD}}$  gets worse. This is probably because when more DoAs are detected, a higher number of wrong frame-wise DoA angles are predicted on average, hence  $LE_{\text{CD}}$  gets worse. “Track-Wise 3” is slightly better than “Track-Wise 2”, which indicates that the mask using SED and EAD predictions is more effective than using SED predictions alone. This additionally proves the significance of EAD.

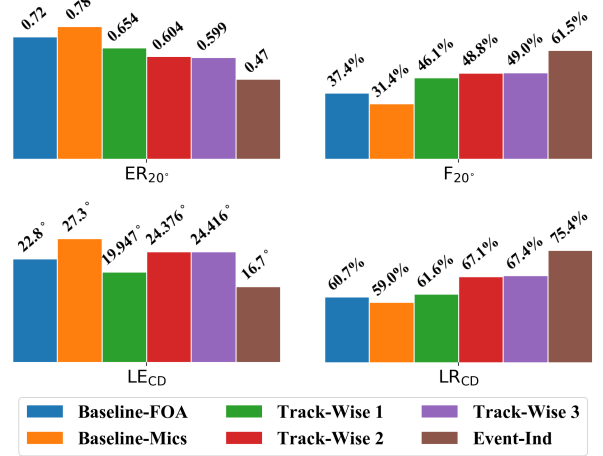


Figure 3: Comparison of different methods.

The “Event-Ind” method achieves the best performance, which means additional EAD and tPIT features all contribute to increase the performance. The additional EAD prediction can constrain and unify predictions from SED and DoA branches both in terms of the temporal information and track-binding. The tPIT can rectify those incorrect label assignments and greatly increase the performance.

## 5. CONCLUSION

We proposed a new end-to-end event-independent network for polyphonic sound event localization and detection. The network treats polyphonic cases as multi-track problems, with each track having at most one event and its corresponding direction-of-arrival. In order to solve the problem of track permutation, a frame-level permutation invariant training strategy is employed. The network outputs three predictions which are sound event detection, event activity detection, and direction-of-arrival. Event activity detection encompasses the feature embedding information from both SED and DoA, hence is able to predict on-set and off-set times of events more accurately. The proposed system is easy to extend to more than two overlapping-event cases. Experimental results show that the proposed system outperforms the baseline methods by a large margin.

## 6. ACKNOWLEDGMENT

This work was supported in part by EPSRC Grants EP/P022529/1, EP/N014111/1 “Making Sense of Sounds”, EP/T019751/1 “AI for Sound”, National Natural Science Foundation of China (Grant No. 11804365), EPSRC grant EP/N509772/1, “DTP 2016-2017 University of Surrey”, and China Scholarship Council (No. 201906470002).

## 7. REFERENCES

- [1] A. Politis, S. Adavanne, and T. Virtanen, "A dataset of reverberant spatial sound scenes with moving sources for sound event localization and detection," *arXiv e-prints: 2006.01919*, 2020.
- [2] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound event localization and detection of overlapping sources using convolutional recurrent neural networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, pp. 34–48, 2018.
- [3] A. Mesaros, S. Adavanne, A. Politis, T. Heittola, and T. Virtanen, "Joint measurement of localization and detection of sound events," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, NY, 2019.
- [4] "DCASE 2020 task 3 sound event localization and detection," 2020. [Online]. Available: <http://dcase.community/challenge2020/task-sound-event-localization-and-detection>
- [5] Y. Cao, Q. Kong, T. Iqbal, F. An, W. Wang, and M. D. Plumbley, "Polyphonic sound event detection and localization using a two-stage strategy," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, NY, 2019, pp. 30–34.
- [6] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 241–245.
- [7] 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 Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1901–1913, 2017.
- [8] T. Virtanen, M. D. Plumbley, and D. Ellis, *Computational Analysis of Sound Scenes and Events*. Springer, 2018, pp. 3–12.
- [9] S. Adavanne, A. Politis, and T. Virtanen, "A multi-room reverberant dataset for sound event localization and detection," in *Submitted to Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, 2019.
- [10] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.
- [11] L. Mazzon, Y. Koizumi, M. Yasuda, and N. Harada, "First order ambisonics domain spatial augmentation for DNN-based direction of arrival estimation," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, NY, 2019, pp. 154–158.
- [12] F. Grondin, I. Sobieraj, M. D. Plumbley, and J. Glass, "Sound event localization and detection using CRNN on pairs of microphones," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, NY, October 2019, pp. 84–88.
- [13] H. Sundar, W. Wang, M. Sun, and C. Wang, "Raw waveform based end-to-end deep convolutional network for spatial localization of multiple acoustic sources," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 4642–4646.
- [14] T. N. T. Nguyen, D. L. Jones, and W.-S. Gan, "A sequence matching network for polyphonic sound event localization and detection," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 71–75.
- [15] Z. Wang, J. Le Roux, and J. R. Hershey, "Multi-channel deep clustering: Discriminative spectral and spatial embeddings for speaker-independent speech separation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 1–5.
- [16] C. Xu, W. Rao, X. Xiao, E. S. Chng, and H. Li, "Single channel speech separation with constrained utterance level permutation invariant training using grid LSTM," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6–10.
- [17] C. Fan, B. Liu, J. Tao, Z. Wen, J. Yi, and Y. Bai, "Utterance-level permutation invariant training with discriminative learning for single channel speech separation," in *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2018, pp. 26–30.
- [18] Y. Liu and D. Wang, "Divide and conquer: A deep CASA approach to talker-independent monaural speaker separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2092–2102, 2019.
- [19] Y. Luo, Z. Chen, and N. Mesgarani, "Speaker-independent speech separation with deep attractor network," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 4, pp. 787–796, 2018.
- [20] Y. Liu, M. Delfarah, and D. Wang, "Deep CASA for talker-independent monaural speech separation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6354–6358.
- [21] 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," *arXiv preprint arXiv:1804.03619*, 2018.
- [22] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 246–250.
- [23] J. Kotus, K. Lopatka, and A. Czyzewski, "Detection and localization of selected acoustic events in acoustic field for smart surveillance applications," *Multimedia Tools and Applications*, vol. 68, no. 1, pp. 5–21, 2014.
- [24] J. Ahonen, V. Pulkki, and T. Lokki, "Teleconference application and B-format microphone array for directional audio coding," in *Audio Engineering Society Conference: 30th International Conference: Intelligent Audio Environments*. Audio Engineering Society, 2007.
- [25] Y. Cao, T. Iqbal, Q. Kong, M. Galindo, W. Wang, and M. D. Plumbley, "Two-stage sound event localization and detection using intensity vector and generalized cross-correlation," *DCASE2019 Challenge*, Tech. Rep., 2019.
- [26] 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.