SOUND EVENT LOCALIZATION AND DETECTION WITH PRE-TRAINED AUDIO SPECTROGRAM TRANSFORMER AND MULTICHANNEL SEPARATION NETWORK

Robin Scheibler, Tatsuya Komatsu, Yusuke Fujita, Michael Hentschel

LINE Corporation, Tokyo, Japan

ABSTRACT

We propose a sound event localization and detection system based on a CNN-Conformer base network. Our main contribution is to evaluate the use of pre-trained elements in this system. First, a pre-trained multichannel separation network allows to separate overlapping events. Second, a fine-tuned self-supervised audio spectrogram transformer provides a priori classification of sound events in the mixture and separated channels. We propose three different architectures combining these extra features into the base network. We first train on the STARSS22 dataset extended by simulation using events from FSD50K and room impulse responses from previous challenges. To bridge the gap between the simulated dataset and the STARSS22 dataset, we fine-tune the models on the training part of the STARSS22 development dataset only before the final evaluation. Experiments reveal that both the pre-trained separation and classification models enhance the final performance, but the extent depends on the adopted network architecture.

Index Terms— SELD, 3D CNN, Conformer, Audio Spectrogram Transformer, Separation

1. INTRODUCTION

Sound event localization and detection (SELD) combines both sound event detection (SED) and direction of arrival (DOA) estimation from multichannel recordings into a single task [1]. The task has been part of the DCASE challenge[1] since 2019. While both tasks are fairly well understood, their combination is made challenging by event polyphony, moving sources, imbalance in the duration of events, interfering events, and an overall training data scarcity. Two types of multichannel recordings have been made available, both derived from 32 channel recordings done with the Eigenmike rigid spherical microphone array[2]. A tetrahydral subset of four channels of the Eigenmike (MIC), and the first order ambisonics (FOA) coefficients derived from all 32 channels. Both formats have four channels and can be used separately or together in the challenge.

Due to the difficulty of the task, all neural solutions have been broadly adopted in the DCASE challenge submissions. A variety of input features have been proposed: generalized cross-correlation, inter-aural level and time differences, intensity vectors [2], per-channel energy normalization [3], SALSA-lite [4], to mention a few. For a detailed list, see [5]. SELD is a data-poor task and augmentations have been a crucial component of past winning systems. One successful strategy is to create new data by convolution of sound event samples with real [6] and simulated [7] impulse responses. To prevent the network overfitting to some directions, the symmetries of the recording system have been used to increase the diversity of DOA angles by artificially rotating the data [8]. A variety of network architectures have been proposed. Many are derived from the convolutional recurrent networks used in the challenge baselines [5]. Many recent solutions have replaced or complemented the recurrent layers by self-attention [9]. The event-independent network architecture [10] proposes to decouple the SED and DOA tasks using separate networks with soft stitching between layers. The 2022 challenge is characterized by the new STARSS22 dataset of real sound scenes played by actors [11]. Due to the high quality of the data set, only 4.9 h of recordings are available. One of the avowed objective of this year's challenge is to explore the use of external ressources to compensate the lack of training data.

Our submission to the DCASE Challenge Task 3 relies on two unrelated advancements in the field. First, high quality pre-trained and self-supervised audio representation have become recently widely available. As examples, see wav2vec [12], PANN [13], and the self supervised audio spectrogram transformer (SSAST) [14]. Second, multichannel source separation based on independent vector analysis (IVA) has been shown to improve sound event detection [15]. Our proposed solution uses the FOA format as it is free of spatial aliasing up to 9 kHz. We use a multichannel separation model trained in advance to coarsely separate the input signal by directions. The separation algorithm is independent vector analysis with a neural source model [16]. Then, we use a pre-trained SSAST [14] fine-tuned on the Task 3 dataset to predict events in the FOA omni channel and the four separation output channels. However, these predictions lack spatial information so we combine them with a dedicated CNN-Conformer network. The inputs of this network are the log-mel spectrograms of the FOA channels and the intensity vector [2]. The CNN creates useful feature maps which are further processed by an eight layer conformer-encoder. We explore three different architectures introducing the SSAST predictions at different points of this network. We find that it is most effective to introduce the SSAST features both at the input and output of the conformer block. The proposed systems are illustrated in Fig. 1.

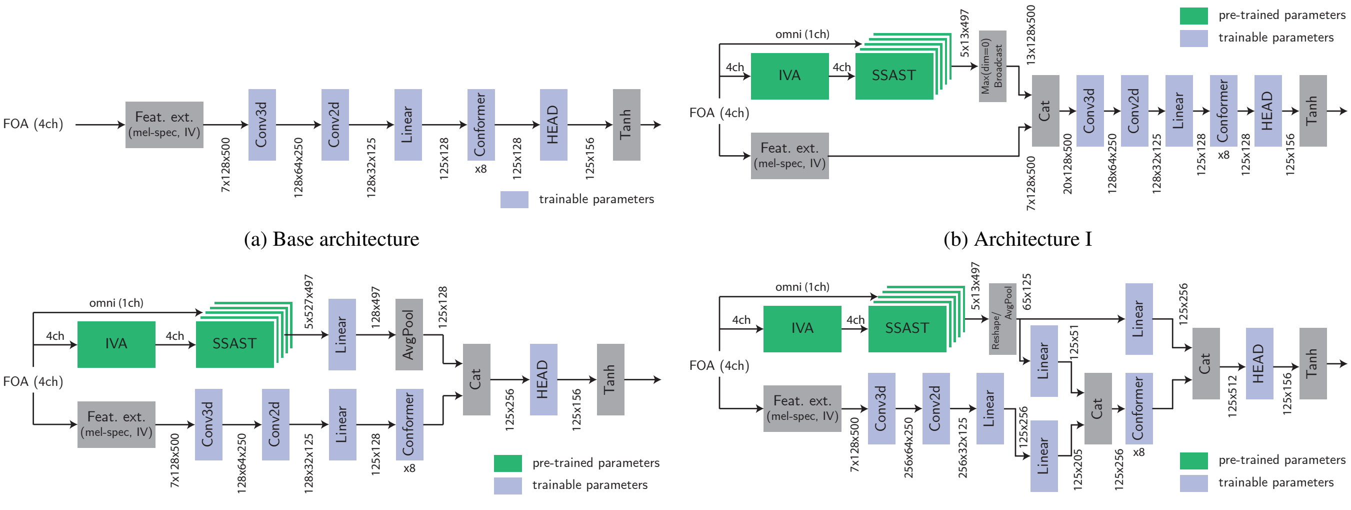
2. PROPOSED SELD NETWORK

2.1. Features

The input data to our SELD network are the four channels first order ambisonics (FOA) signals. First, to help with recognition of events, we run the FOA into a separation network that roughly separates the different events. The separation network is based on independent vector analysis [17] with a neural source model [16] described in Section 2.1.1. We obtain four tracks out of the separation network. Second, these four tracks as well as the omni channel of the FOA are run through a fine-tuned Self-Supervised Audio Spectrogram

[1] https://dcase.community

[2] https://mhacoustics.com/products



(a) Base architecture

(b) Architecture I

(c) Architecture II

(d) Architecture III

Figure 1: Structure of the proposed systems. Blue blocks have trainable parameters. Green blocks have been pre-trained. Gray blocks are not trainable. The (a) base system and the three extensions proposed (b), (c), and (d). HEAD is either a linear or MLP layer.

Transformer (SSAST) described in Section 2.1.2. Third, the log-mel-spectrograms of the four FOA channels as well as the intensity vector (IV) [2], as used in many SELD systems, provide the spatial information needed for DOA estimation. We use 128 bands for the mel-spectrogram analysis.

2.1.1. Separation Network

The multichannel separation network consists of a blind dereverberation part using weighted prediction error (WPE) [18], followed by independent vector analysis (IVA) [19, 20]. For WPE, the STFT uses an FFT size of 512 with ¾-overlap and a Hann window. The number of iterations, delays, and taps is 3, 3, and 10, respectively. For IVA, the STFT uses an FFT size of 2048 with ¾-overlap and a Hann window. The IVA algorithm used is iterative source steering [17] with a neural source model [16]. The number of IVA iterations is 20 and we use demixing matrix checkpointing [21] to save memory. The neural source model uses three 1D convolutional layers with GLU non-linearities and batch normalization with four groups. The hidden dimension is 128 which we map back to the STFT size by a 1D transposed convolution layer. Finally, a sigmoid non-linearity produces a mask-like signal from the network's output. A system description of the IVA separation and neural source models are shown in Fig. 2.

Since we do not have access to the ground-truth separated signals for the SELD datasets, we cannot use the conventional source separation loss functions, e.g., SI-SDR or CI-SDR. However, we have access to the direction of arrival of the events so that we can use a recently proposed spatial loss [22]. To train the network, we cut the input data into blocks of 5 s and use the median DOA as target because IVA assumes the sources to be static in this interval.

2.1.2. Self-Supervised Audio Spectrogram Transformer

The Self-Supervised Audio Spectrogram Transformer (SSAST) [14] is an all-attention model that has been extensively pre-trained by self-supervision on Audioset [23]. We fine-tune a pre-trained version of SSAST [24] on the STARSS22 dataset and the baseline extended dataset prepared by the organizers of Task 3. The fine tuning is done for the SED part of the task only. To this end, the DOA information is stripped from the targets and multiple events of the same class are merged together when they appear simultaneously. The SSAST model operates on 5 s blocks and produces class presence prediction vectors (13-dimensional) for each of the 497 frames (approx. 10 ms per frame).

2.2. SELD Network

Our proposed SELD system combines a base network with the extra predictions obtained from the separation network and the SSAST. The different system variants are shown in Fig. 1 and their number of parameters given in Table 1.

Base Network: The base network is a fairly conventional CNN-Conformer network for SELD using FOA features. We feed the log-mel-spectrograms of the four FOA channels and the IV channels into a convolutional network with two layers (total of 7 channels). The first is a 3D convolutional layer where the three dimensions are channels, mel-frequency bands, and time, respectively. We expect that such 3D filters can better capture the directional information present in the input signal. The kernels are of size $7 \times 3 \times 3$ and the padding is $(0, 1, 1)$, which results in a 2D output signal. Thus, the second layer is a 2D convolutional layer with 3×3 kernels. Strides of size 2 are used in the frequency and time dimension to reduce the size of the input signal. The number of channels after the 3D convolution is 128. Group normalization with four groups and ReLU activations are used after each layer. After the two strided convolutions, the remaining 32 frequency dimensions are merged with the 128 channels and projected to dimension 128 by a linear layer before the output. The output of this stage is an embedding signal with 128 dimensions and a frame interval of 40 ms This output is fed into a conformer-encoder [25] with eight layers and convolution kernel size 7. The embedding vectors so created are then projected by an output head. We explore both a simple linear projection and a two layer multi-layer perceptron (MLP). The MLP uses a GeLU non-

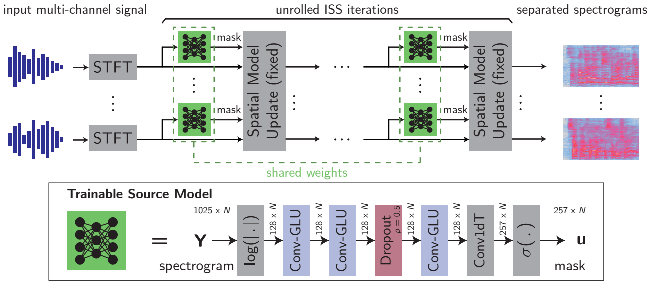


Figure 2: The structure of the separation network used to obtain the input features.

Model	SSAST	IVA	Base	I	II	III
Params.	87.2 M	2.4 M	3.8 M	3.8 M	3.9 M	15.3 M

Table 1: List of models and their number of parameters. Numbers reported include the MLP output head.

linearity. The final non-linearity is a hyperbolic tangent to limit the output to the $[-1, 1]$ range. The output is in the Multi-ACCDOA format [26] with 4 tracks, thus the output size is 4 tracks × 3 dimension of Cartesian DOA vectors × 13 classes, a total of 156 outputs per time frame. The event presence probability is given by the length of the 3D vector for each track/class slot. The base network is shown in Fig. 1a.

Architecture I: Our first variant combines all the features at the input of the CNN (Fig. 1b). The 5 channels of SSAST predictions are aggregated by taking the max. Then, they are broadcasted to match the 128 bands of the log-mel spectrogram and IV features to which they are concatenated. Thus, the input of the CNN has 20 channels and 128 mel-frequency bands. The rest of the network is similar to the base network.

Architecture II: Here we concatenate the SSAST predictions to the output of the conformer, before the output head (Fig. 1c). We project the SSAST prediction vectors of the omni FOA channel and the 4 IVA output channels (see Section 2.1.1) from 13 to 128 dimensions by a linear projection followed by ReLU activations. After this, these five channels are averaged into one. The frame rate is adjusted to that of the spatial feature extraction network by average pooling of size four along the time axis. The embedding obtained is concatenated to the output of the conformer to obtain an embedding of size 256. Finally, a linear layer projects this concatenated embedding to the output size.

Architecture III: This variant combines the SSAST predictions to both the input and the output of the conformer (Fig. 1d). The 5 channels and 13 classes dimensions are reshaped to size 65 and projected to size 51 by a linear layer. The output of the CNN is linearly projected to size 205 and concatenated to obtain an embedding of size 256 which is fed to the conformer-encoder. Another linear layer projects the 65-dimensional pre-trained features to size 256, which is concatenated to the output of the conformer layers to obtain a 512-dimensional embedding fed to the output head. This variant uses an embedding size of 256 both for the CNN and conformer (unlike 128 for all other networks), resulting in a larger network.

Name	Ref	Type	Ov.	Inter.	Train	Val.
STARSS22	[11]	Rec.	5	✓	2.9 h	2.0 h
Synth1	[27]	Sim.	2	0	20 h	—
Synth2		Sim.	4	1	20 h	—

Table 2: The datasets used. Columns “Ov.” and “Inter.” indicate the maximum number of overlapping event, and the number of interfering out-of-classes events. Rec. and Sim. stand for “recorded” and “simulated”, respectively.

Name	Ref.	Type
STARSS22	[11]	DCASE2022 task 3 dataset
FSD50K	[28, 29]	audio dataset
TAU-SRIR DB	[6, 30]	RIR dataset
SSAST	[14, 24]	pre-trained pytorch model

Table 3: List of external datasets and models used

2.3. Post-processing

The post-processing works in two steps. Let $\boldsymbol{q}_{ntc}$ be the output of the nth frame, tth track, and cth class. The event probability is taken to be $p_{ntc} = \|\boldsymbol{q}_{ntc}\|$. First, events are detected if $p_{ntc} \geq \sigma_c$ at the output framerate of the network. We run a de-duplication procedure to remove duplicate events produced by the Multi-ACCDOA format. Events from different tracks of the same class with directions closer than θ_c, a class specific threshold, are merged together. Second, all the events from the same output frame are aggregated together. Because the output frames of the network are $40\,\mathrm{ms}$ and the target frames are $100\,\mathrm{ms}$, there are 2 or 3 events per output frame, track, and class. For every output frame and class, we find the event with largest p_{ntc} and count all events within θ_c. If the count is larger than η_c, we declare an event with direction given by the average of all aggregated events, weighted by their probability. By default, we use $\sigma_c = 1/2$, $\theta_c = 15°$, and $\eta_c = 1$. To maximize performance, we use a post-processing calibration procedure where σ_c, θ_c, and η_c are chosen per class to minimize the SELD score on the validation dataset of STARSS22.

2.4. Differences with the Challenge Submission

Our challenge submission was based on Architecture II with a linear output head. However, after the end of the challenge, we found a mistake in our use of the pre-trained SSAST, namely, the output layers were not initialized to the correct size (527 instead of 13 dimensional vectors) and weights. Furthermore the weights used on the development and evaluation datasets were different. We have corrected and retrained all the architectures for this paper.

3. DATASET AND TRAINING

3.1. Datasets

We use the three datasets described in Table 2 with a total of $42.9\,\mathrm{h}$ and $2.0\,\mathrm{h}$ of training and validation data, respectively. From the DCASE2022 task 3 dataset, STARSS22 [11], fold3 ($2.9\,\mathrm{h}$) is used for training and fold4 ($2.0\,\mathrm{h}$) for validation, as suggested. Since this is not sufficient, we also use the baseline training synthetic dataset

(Synth1) provided by the task organizers [27]. This dataset is created by remixing sound events from the FSD50K dataset [28, 29] with the measured RIR from the TAU-SRIR database [30, 6]. However, the dataset Synth1 only contains up to two overlapping events, and no interfering events. Thus, we use the original recipe provided for Synth1 [31] to create an extended training set, Synth2. We change the recipe in the following ways. First, increase the maximum number of overlapping events from 2 to 4. Second, we add interfering sound events not included in the classification task. For the interference, we select clips from the following categories of FSD50K: `Cutlery, silverware`, `Computer, keyboard`, `Chewing, mastication`, `Buzz`, `Crumpling, crinkling`, `Typing`, `Clock`, `Meow`, `Breathing`, `Glass`, `Writing`, `Chink, clink`. The base external datasets and pretrained models used are summarized in Table 3 and the training datasets in Table 2, respectively.

3.2. Data Augmentations

SpecAugment: We apply SpecAugment [32] using the same mask to all FOA channels prior to computation of mel-spectrogram and IV during training. The maximum time masking is 2 % of the total length, while frequency masking is up to 10 %.

Random Rotations: To avoid the network over-fitting to specific directions, we apply random rotations to the FOA input, as has been successfully used for SELD networks in previous challenges [8]. By applying the same rotation to the targets, we are able to simulate large spatial variations in the input dataset. This augmentation is applied to input examples with probability ½.

3.3. Training

We train the network with the recently proposed Multi-ACCDOA loss [26]. The optimizer is Adam [33] with learning rate 0.001. We do learning rate warm-up over the first 10000 steps. The network is trained for 1000 epochs on STARSS22, Synth1, and Synth2 datasets. The progress of the optimization is monitored on the validation set of STARSS22 using the SELD score,

$$\mathrm{SELD} = 0.25\,(\mathrm{ER} + (1 - \mathrm{F}) + \mathrm{LE}\,/180 + (1 - \mathrm{LR}))\,, \quad (1)$$

where ER, F, LE, LR, are the official SELD metrics [1]. After training finishes, we fine-tune the network on the training part (fold3) of STARSS22 only. We freeze all layers except the output head. Training is restarted from the average of the 10 checkpoints with lowest SELD score with learning rate 0.0001. We proceed for 1000 epochs. Finally, we select the 10 checkpoints with the lowest validation score and average their weights.

4. EXPERIMENTS

We do an ablation study to assess the contribution of the different components. Our reference is the base network with linear output head, trained without fine tuning (Fig. 1a). For each architecture we add in order SSAST predictions of FOA omni channel (+AST), and four separation outputs (+IVA), MLP output head (+MLP), fine-tuning (+FINE), and post-processing calibration (+POST).

Table 4 shows the results on the validation part of the development set of STARSS22 (fold4) compared to that of the baseline system [34]. Our first observation is that all the proposed architectures improve significantly over the baseline [34]. In addition, we see that how we insert the SSAST/IVA features into the network matters. Architecture I concatenates the extra features at the input of the network, which might make training the entire network more difficult. The MLP head is important here to obtain the best performance. For architecture II, where the features are only inserted at the end, the MLP is also required to fully take advantage of the extra information. Architecture III performs best by considering features both in the conformer, and again in the output head. There, the benefits of SSAST and IVA features are clearly visible. Of the three, it performs best, but is also the largest model. However, the better performance is not due to size only since it does not outperform I and II for +AST. In all cases, fine-tuning and post-processing calibration are necessary to maximize performance.

Model	ER↓	F↑	LE↓	LR↑	SELD ↓
Baseline (FOA) [34]					
	0.71	0.21	29.3	0.46	0.5507
Base Network					
	0.578	0.421	19.083	0.602	0.4154
+MLP	0.594	0.412	17.015	0.608	0.4174
+FINE	0.561	0.451	16.314	0.563	0.4094
+POST	0.535	0.464	**15.869**	0.562	0.3994
Architecture I					
+AST	0.575	0.423	18.752	0.591	0.4164
+IVA	0.574	0.418	17.809	0.582	0.4182
+MLP	0.584	0.455	17.331	0.606	0.4050
+FINE	0.562	0.469	16.881	0.616	0.3928
+POST	0.519	0.480	16.375	0.598	0.3830
Architecture II					
+AST	0.572	0.424	18.130	0.604	0.4111
+IVA	0.589	0.414	18.016	0.611	0.4160
+MLP	0.592	0.445	18.156	**0.641**	0.4020
+FINE	0.534	0.478	17.163	0.595	0.3891
+POST	0.516	0.497	16.551	0.603	0.3768
Architecture III					
+AST	0.579	0.417	18.785	0.607	0.4147
+IVA	0.572	0.437	17.957	0.621	0.4037
+MLP	0.567	0.460	18.294	0.616	0.3980
+FINE	0.551	0.493	17.505	0.639	0.3792
+POST	**0.500**	**0.514**	17.131	0.624	**0.3644**

Table 4: SELD metrics of proposed architectures and baseline [34].

5. CONCLUSION

We have presented three ways of using a pre-trained SSAST and separation system to improve SELD. An ablation study demonstrated the effectivness of the different components. We found that inserting the pre-trained predictions both before and after the conformer encoder, combined with an MLP outptut classification head is most effective.

6. REFERENCES

[1] A. Politis, A. Mesaros, S. Adavanne, T. Heittola, and T. Virtanen, “Overview and evaluation of sound event localization and detection in dcase 2019,” *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 29, pp. 684–698, 2020.

[2] K. Lopatka, J. Kotus, and A. Czyzewski, "Detection, classification and localization of acoustic events in the presence of background noise for acoustic surveillance of hazardous situations," *Multimedia Tools and Applications*, vol. 75, no. 17, pp. 10 407–10 439, 2016.

[3] V. Lostanlen et al., "Per-Channel Energy Normalization: Why and How," *IEEE Signal Process. Lett.*, vol. 26, no. 1, pp. 39–43, Jan. 2019.

[4] T. N. Tho Nguyen, D. L. Jones, K. N. Watcharasupat, H. Phan, and W.-S. Gan, "SALSA-Lite: A fast and effective feature for polyphonic sound event localization and detection with microphone arrays," in *Proc. IEEE ICASSP*, Singapore, SG, May 2022.

[5] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound event localization and detection of overlapping sources using convolutional recurrent neural networks," *IEEE J. Sel. Top. Signal Process.*, vol. 13, no. 1, pp. 34–48, Apr. 2019.

[6] A. Politis, S. Adavanne, and T. Virtanen, "A dataset of reverberant spatial sound scenes with moving sources for sound event localization and detection," in *Proc. DCASE*, Tokyo, JP, Nov. 2020.

[7] Y. Koyama et al., "Spatial data augmentation with simulated room impulse responses for sound event localization and detection," in *Proc. IEEE ICASSP*, Singapore, SG, May 2022, pp. 8872–8876.

[8] F. Ronchini, D. Arteaga, and A. Pérez-López, "Sound event localization and detection based on crnn using rectangular filters and channel rotation data augmentation," in *Proc. DCASE2020*, Tokyo, JP, Nov. 2020.

[9] K. Shimada et al., "Ensemble of ACCDOA- and EINV2-based Systems with D3Nets and Impulse Response Simulation for Sound Event Localization and Detection," DCASE Challenge, Tech. Rep., June 2021.

[10] Y. Cao, T. Iqbal, Q. Kong, F. An, W. Wang, and M. D. Plumbley, "An Improved Event-Independent Network for Polyphonic Sound Event Localization and Detection," in *Proc. IEEE ICASSP*, Toronto, CA, June 2021, pp. 885–889.

[11] A. Politis et al., "STARSS22: A dataset of spatial recordings of real scenes with spatiotemporal annotations of sound events," *arXiv preprint arXiv:2206.01948*, 2022.

[12] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Proc. Interspeech*, Graz, AU, Sept. 2019, pp. 3465–3469.

[13] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 2880–2894, Oct. 2020.

[14] Y. Gong, C.-I. J. Lai, Y.-A. Chung, and J. Glass, "SSAST: Self-supervised audio spectrogram transformer," *arXiv preprint arXiv:2110.09784*, 2021.

[15] R. Scheibler, T. Komatsu, and M. Togami, "Multichannel separation and classification of sound events," in *Proc. EUSIPCO*, Dublin, IE, Aug. 2021, pp. 1035–1039.

[16] R. Scheibler and M. Togami, "Surrogate source model learning for determined source separation," in *Proc. IEEE ICASSP*, Toronto, CA, June 2021, pp. 176–180.

[17] R. Scheibler and N. Ono, "Fast and stable blind source separation with rank-1 updates," in *Proc. IEEE ICASSP*, Barcelona, ES, May 2020, pp. 236–240.

[18] T. Yoshioka and T. Nakatani, "Generalization of Multi-Channel Linear Prediction Methods for Blind MIMO Impulse Response Shortening," *IEEE Trans. Audio Speech Lang. Process.*, vol. 20, no. 10, pp. 2707–2720, 2012.

[19] A. Hiroe, "Solution of permutation problem in frequency domain ica, using multivariate probability density functions," in *ASIACRYPT 2016*. Springer Berlin Heidelberg, Jan. 2006, vol. 3889, pp. 601–608.

[20] T. Kim, T. Eltoft, and T.-W. Lee, "Independent vector analysis: An extension of ICA to multivariate components," in *ASIACRYPT 2016*. Springer Berlin Heidelberg, Jan. 2006, vol. 3889, pp. 165–172.

[21] K. Saijo and R. Scheibler, "Independence-based joint dereverberation and separation with neural source model," in *Proc. Interspeech*, Incheon, KR, Sept. 2022.

[22] ——, "Spatial loss for unsupervised multi-channel source separation," in *Proc. Interspeech*, Incheon, KR, Sept. 2022.

[23] J. F. Gemmeke et al., "Audio Set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE ICASSP*, New Orleans, LA, USA, Mar. 2017, pp. 776–780.

[24] Y. Gong et al., "Yuangongnd/ssast." [Online]. Available: https://github.com/YuanGongND/ssast

[25] A. Gulati et al., "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, 2020, pp. 5036–5040.

[26] K. Shimada et al., "MULTI-ACCDOA: Localizing and detecting overlapping sounds from the same class with auxiliary duplicating permutation invariant training," in *Proc. IEEE ICASSP*, Singapore, SG, pp. 316–320.

[27] A. Politis, "[DCASE2022 Task 3] Synthetic SELD mixtures for baseline training," Apr. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6406873

[28] E. Fonseca et al., "FSD50K: an open dataset of human-labeled sound events," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 30, pp. 829–852, 2022.

[29] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K," Oct. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4060432

[30] A. Politis, S. Adavanne, and T. Virtanen, "TAU Spatial Room Impulse Response Database (TAU- SRIR DB)," Apr. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6408611

[31] D. Krause and A. Politis, "danielkrause/dcase2022-data-generator." [Online]. Available: https://github.com/danielkrause/DCASE2022-data-generator

[32] D. S. Park et al., "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*. ISCA, Sept. 2019, pp. 2613–2617.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, San Diego, CA, USA, May 2015.

[34] S. Adavanne, "sharathadavanne/seld-dcase2022." [Online]. Available: https://github.com/sharathadavanne/seld-dcase2022