3D CNN AND CONFORMER WITH AUDIO SPECTROGRAM TRANSFORMER FOR SOUND EVENT DETECTION AND LOCALIZATION

Technical Report

Robin Scheibler, Tatsuya Komatsu, Yusuke Fujita, Michael Hentschel

LINE Corporation, Tokyo, Japan

ABSTRACT

We propose a network for sound event detection and localization based on a 3D CNN for the extraction of spatial features followed by several conformer layers. The CNN performs spatial feature extraction and the subsequent conformer layers predict the events and their locations. We combine this with features obtained from a fine-tuned audio-spectrogram transformer and a multi-channel separation network trained separately. The two architectures are combined by a linear layer before the final non-linearity. We first train the network 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 model on the development part of the STARSS22 dataset only before the final evaluation.

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

1. INTRODUCTION

We propose a solution to simultaneously classify sound events and estimate their location. Our solution mixes the use of a pre-trained self-supervised model, a pre-trained multi-channel separation model, and a dedicated network for sound event detection and localization (SELD). The self-supervised model is the self-supervised audio spectrogram transformer (SSAST) [1]. The multichannel separation algorithm is independent vector analysis with a neural source model [2]. We focus on the first order ambisonics (FOA) signals as they do not contain spatial aliasing up to 9 kHz. We use the separation algorithm and SSAST to produce high quality features for sound event detection. However, both features do not contain much spatial information so we combine them with dedicated SELD network. The SELD network is composed of a 3D CNN as input which allows to process the input channels jointly. The features produced by the CNN are then processed by an eight layer conformer encoder. The concatenation with the SSAST features is then projected by a linear layer to obtain the final output. A diagram of the whole system is provided in Fig. 1.

2. PROPOSED SELD NETWORK

2.1. Features

The input data to our network are the 4 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 [3] with a neural source model [2] described in Section 2.1.1. We obtain four tracks out of the separation network. These four tracks as well as the omni channel of the FOA are run through a fine-tune Self-Supervised Audio Spectrogram Transformer (SSAST) described in Section 2.1.2. The spatial features are the log-mel-spectrograms of the four FOA channels as well as the intensity vectors (IV) [4] as used in previous SELD systems. We use 128 bands for the mel-spectrogram analysis.

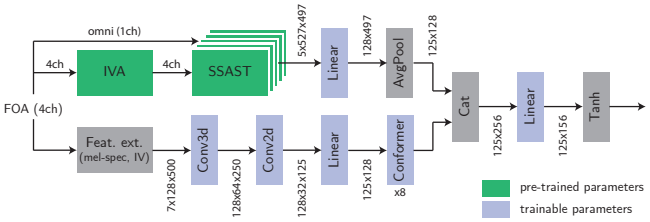

Figure 1: Structure of the proposed systems. Blue blocks have trainable parameters. Greene blocks have been pre-trained. Gray blocks are not trainable.

2.1.1. Separation Network

The multichannel separation network consists of a blind dereverberation part using weighted prediction error (WPE) [5, 6], followed by independent vector analysis (IVA) [7, 8]. For WPE, the STFT uses 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 FFT size of 2048 with ¾ overlap and a Hann window. The IVA algorithm used is iterative source steering [3] with a neural source model [2]. The number of IVA iterations is 20 and we use demixing matrix checkpointing [9] to save memory. The neural source model uses three 1D convolutional layers with GLU non-linearities and group 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 [10]. To train the network, we cut the input data into 5 s blocks. Since IVA assumes the sources to be static in this interval, we use the median DOA as target.

2.1.2. *Self-Supervised Audio Spectrogram Transformer*

The Self-Supervised Audio Spectrogram Transformer (SSAST) [1] is an all-attention model that has been extensively pre-trained by self-supervision on Audioset. We fine-tune a pre-trained version of SSAST [11] 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 merged together when they appear simultaneously. The SSAST model operates on 5 s blocks and produces 527-dimensional embedding vectors for each of the 497 frames (approx. 10 ms per frame).

2.2. SELD Network

The main purpose of our network is to extract spatial information from the input 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. The three dimensions are channels, mel-frequency bands, and time. We expect that such 3D filters will be better able to capture the directional information present in the input signal. The kernels are of size $7 \times 3 \times 3$. The padding used 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. 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 [12] with eight layers and convolution kernel size 7.

WWe project the SSAST embedding vectors of the omni FOA channel and the 4 IVA output channels (see Section 2.1.1) from 527 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. The output is in the Multi-ACCDOA format [13] with 4 tracks, thus the output size is 4 tracks $\times$ 3 dimension of Cartesian DOA vectors $\times$ 13 classes, a total of 156 outputs per time frame. We use a hyperbolic tangent non-linearity to limit the output to the $[-1, 1]$ range. The event presence probability is given by the length of the 3D vector for each track/class slot. The whole system is illustrated in Fig. 1. The number of parameters of the different modules are given in Table 1.

2.3. Post-processing

The post-processing works in two steps. For the explanation, 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 deduplication procedure to remove duplicate events produced by the Multi-ACCDOA. Events from different tracks of the same class with directions closer than θ_c, a class specific threshold, are merged together. Second, we aggregate all the events from the same output frame together. The output frames of the network are 40 ms and the target frames are 100 ms. We have 2 or 3 events per output frame, track, and class (at most). 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 strictly larger than θ_c, we declare an event with direction being the average of all aggregated events, weighted by their probability.

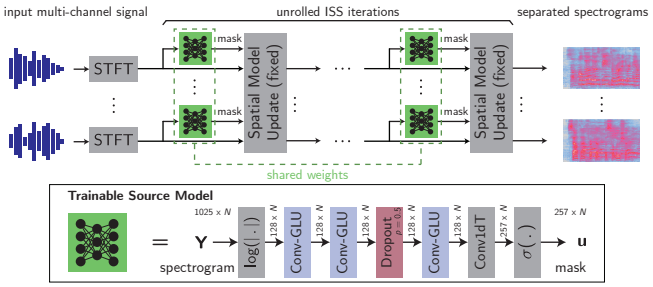

Figure 2: The structure of the separation network used to obtain the input features.

Model	# Parameters
SSAST	87582943
IVA	2366849
SELD	3861020
Total	93810812

Table 1: List of models used and their number of parameters.

3. DATASET AND TRAINING

3.1. Datasets

We use the three datasets described in Table 3 with a total of 42.9 h and 2.0 h of training and validation data, respectively. From the DCASE2022 task 3 dataset, STARSS22 [14], fold3 (2.9 h) is used for training and fold4 (2.0 h) for validation, as suggested. Since this is not sufficient, we also use the baseline training synthetic dataset (Synth1) provided by the task organizers [19]. This dataset is created by remixing sound events from the FSD50K dataset [15, 16] with the measured RIR from the TAU-SRIR database [18, 17]. However, the dataset Synth1 only contains up to two overlapping events, and no interfering events. Thus, we use the original recipe provided to construct Synth1 [20] to create an extended training set, Synth2. We change the recipe in the following ways.

1. Increase the maximum number of overlapping events from 2 to 4.

Name	Ref.	Type
STARSS22	[14]	DCASE2022 task 3 dataset
FSD50K	[15, 16]	audio dataset
TAU-SRIR DB	[17, 18]	RIR dataset
SSAST	[1, 11]	pre-trained pytorch model

Table 2: List datasets and models used

Name	Ref	Type	Ov.	Inter.	Train	Val.
STARSS22	[14]	Rec.	5	✓	2.9 h	2.0 h
Synth1	[19]	Sim.	2	0	20 h	—
Synth2		Sim.	4	1	20 h	—

Table 3: The datasets used. Columns "Ov." and "Inter." indicate the maximum number of overlapping event, and the number of interfering out-of-classes events. Synth1 was provided by the task organizers. Synth2 was created by the authors based on the recipe provided to create synth1 [20]. For the validation, the test set of STARSS22 was used.

2. 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 pre-trained models used are summarized in Table 2 and the training datasets in Table 3, respectively.

3.2. Data Augmentations

SpecAugment We apply SpecAugment [21] using the same mask to all FOA channels prior to computation of mel-spetrogram and intensity vector 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 locations, we apply random rotations to the FOA input, as has been successfully used for SELD networks in previous challenges [22]. 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 [13]. The optimizer is Adam [23] with learning rate 0.001. 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,

$$\mathrm{SELD} = \frac{1}{4}\left(\mathrm{ER} + (1-\mathrm{F}) + \frac{\mathrm{LE}}{180} + (1-\mathrm{LR})\right), \qquad (1)$$

where ER, F, LE, LR, are the official SELD metrics [24]. After training finishes, we fine-tune the network on the training part of STARSS22 only, restarting from the checkpoint with lowest SELD score with learning rate 0.0002. We proceed until the validation SELD score starts increasing again. Finally, we select the 10 checkpoints with lowest validation score and average their weights.

4. RESULTS

Table 4 shows our results on the validation set of STARSS22 compared to that of the baseline system [25].

Model	Input	ER	F	LE	LR
Baseline	MIC	0.71	0.18	32.2	0.47
Baseline	FOA	0.71	0.21	29.3	0.46
Proposed	FOA	0.51	0.49	16.9	0.63

Table 4: SELD metrics of the proposed system compared to that of the baseline system [25].

5. REFERENCES

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

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

[3] 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.

[4] 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.

[5] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. Juang, "Speech dereverberation based on variance-normalized delayed linear prediction," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 7, pp. 1717–1731, Sept. 2010.

[6] 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.

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

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

[9] K. Saijo and R. Scheibler, "Independence-based joint dereverberation and separation with neural source model," *arXiv preprint arXiv:2110.06545*, 2021.

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

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

[12] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, 2020, pp. 5036–5040.

[13] K. Shimada, Y. Koyama, S. Takahashi, N. Takahashi, E. Tsunoo, and Y. Mitsufuji, "MULTI-ACCDOA: Localizing and detecting overlapping sounds from the same class with auxiliary duplicating permutation invariant training," in *Proc. IEEE ICASSP*, Singapore, Singapore, pp. 316–320.

[14] A. Politis, K. Shimada, P. Sudarsanam, S. Adavanne, D. Krause, Y. Koyama, N. Takahashi, S. Takahashi, Y. Mitsufuji, and T. Virtanen, "STARSS22: A dataset of spatial recordings of real scenes with spatiotemporal annotations of sound events," *arXiv preprint arXiv:2206.01948*, 2022.

[15] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: an open dataset of human-labeled sound events," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 30, pp. 829–852, 2022.

[16] ——, "FSD50K," Oct. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4060432

[17] 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.

[18] ——, "TAU Spatial Room Impulse Response Database (TAU- SRIR DB)," Apr. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6408611

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

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

[21] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*. ISCA, Sept. 2019, pp. 2613–2617.

[22] 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.

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

[24] 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.

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