Sound Event Localization and Detection Evaluated in Real Spatial Sound Scenes

Blaž Erzar

1 Introduction

The DCASE2023 Challenge [Shi+23] consisted of multiple tasks revolving around the detection and classification of acoustic scenes and events in audio recordings. For this project, we delved into *Task 3, Sound Event Localization and Detection Evaluated in Real Spatial Sound Scenes*. We checked existing work, reproduced the baseline system implementation, and evaluated the baseline model as well as an improved model architecture in different configurations.

This task is a continuation of the same task from the DCASE2022 Challenge [Pol+22]. The data has been expanded with additional recordings, which also include simultaneous 360° video recordings [Pol+23]. Because of this, the task has been split into two tracks, *Track A: Audio-only inference* and *Track B: Audiovisual inference*. In this project, we focus on the former and only deal with audio.

The dataset has been collected in Tampere, Finland, by the Audio Research Group of Tampere University, and in Tokyo, Japan, by Sony. The dataset contains the development and evaluation sets, but we only work on the development set and its splits because we do not have access to the evaluation set annotations. The development set was recorded in multiple rooms and split into a train (4.2 h) and test split (3.2 h).

The audio was recorded using two microphone setups. One was using an Eigenmike that records the *first-order ambisonics (FOA)*, and the other one consisted of four microphones in a special arrangement to record the *tetrahedral microphone array (MIC)*. In our implementation, we used the FOA recordings, as has been the case with most of the teams that participated in this challenge. This format encodes the sound field into four channels, one omnidirectional (W), and three bidirectional (X, Y, Z) [Art25].

Each recording was annotated on a 100 ms resolution, so-called *frames*. There are 13 target sound event classes. In each frame, multiple events can occur simultaneously, even of the same class. Over the whole dataset, approximately 85% of frames were annotated (contained at least one event), 36% contained multiple events, and 10% contained multiple events of the same class. Each frame annotation contains information of the *active class*, *source index* (to distinguish sources in the scene), *azimuth*, *elevation* and *distance*. The azimuth and elevation angles represent the direction of the audio source and are the main goals of our prediction. The source index and distance are only there as additional information.

2 Related work

The goal of *Sound Event Detection (SED)* systems is to automatically recognise what is happening and when [Mes+21]. In the most basic setting, only a single sound is active at once. If multiple sounds can be active in parallel, we are dealing with *Polyphonic SED*. In both cases, sound categories need to be defined in some way. Because there are different ways of defining categories of sounds, ambiguity can be problematic. All systems are capable of detecting the predefined target classes, contrary to speech recognition, where a system is capable of detecting all combinations of sounds.

In this challenge, the problem is expanded to *Polyphonic Sound Event Detection and Localization (SELD)*, as well as instance-aware detection [Pol+20a], since there can be multiple instances of the same class. At first, the problems of detection and localization were treated separately, but with the development of deep neural networks, they became the most established method for solving both of these tasks simultaneously.

The first solutions were based on the CNN architecture, with later ones expanding to a convolutional and recurrent neural network (CRNN). The same architecture was also used by the baseline implementation for the DCASE2022 Challenge [Ada+19]. Next year's baseline implementation, which we implemented, expanded this model with multi-head attention layers after the recurrent layers.

The latest model used by many participants is the Conformer [Gul+20]. It combines the convolutional layers with a transformer-like architecture. The model is much larger than the baseline model, but it performed best in the 2023 execution of the competition. We implemented this model to compare it to the baseline implementation.

3 Methods

3.1 Features

When working with audio, the most commonly used feature is the *log-mel spectrogram*. It is computed using the short-time Fourier transform. We use 1024 for the FFT size, 40 ms for the window size, and 20 ms for the hop length between STFT windows. We use 64 filterbanks and the power spectrogram, to get to the mel scale, which is a perceptual scale of pitches. Finally, the amplitude scale is transformed to the decibel scale to get the log-mel spectrogram $S \in \mathbb{R}^{64\times 5T}$,

where T is the number of frames in the recording. We get a separate spectrogram for each audio channel.

Because we are using FOA audio format, we also use the *intensity vector* [Yas+20] as an additional feature. Let W, X, Y, Z be the outputs of the STFT. The intensity vector is defined as:

$$\mathbf{I} \propto \Re(W^*[X, Y, Z]^\mathsf{T}),$$

and normalised to unit length. To get the same dimensions as in the spectrogram, we use the same filterbanks to transform the intensity vector. Concatenating all features together, we get a matrix of size $7 \times 64 \times 5T$ for each recording.

3.2 Sound event detection (SED)

First, we tackled a simpler problem of detection by implementing the baseline SELDnet model [Ada+19]. We use 3 convolutional layers with kernel size 3. Each one is followed by batch normalisation, ReLU, maximum pooling and dropout. For pooling, we use sizes $(4, 5)$, $(4, 1)$, and $(2, 1)$. This makes sure we get a single output for each time frame. The convolution blocks are followed by 2 layers of bidirectional GRU with dimension 128 and $\tanh$ activation function, and 2 layers of multi-head attention with 8 heads followed by layer normalisation. As in the original implementation, we use 0.05 for dropout probability.

The above model represents only the backbone of our model, which we later also use for localization. To get the correct output, we add a single linear layer that outputs $C \cdot E$ values, where C is the number of classes (13) and E the maximum number of simultaneous events (5).

We trained the model using two loss functions: softmax and sigmoid. With softmax, we output a single class for each event. Because of this, we need to have an additional class for a non-event. With sigmoid, we classify each class and event combination independently. In this case, no additional class is needed.

3.3 Data augmentation

As is evident from technical reports from the competition, all teams used some sort of data augmentation or additional datasets. In this domain, there is not a lot of labelled data available, and till 2021, they were still using synthetic datasets.

To increase the amount of data, we use the augmentations proposed in [Wan+23]. Because we have FOA recordings, we can manipulate the audio channels to obtain a *rotated* or *flipped* recording. E.g., computing new audio channels as $C_1^{\text{new}} = C_1$, $C_2^{\text{new}} = -C_4$, $C_3^{\text{new}} = -C_3$, and $C_4^{\text{new}} = C_2$, we get a recording with labels $\phi^{\text{new}} = \phi - \pi/2$ and $\theta^{\text{new}} = -\theta$, where ϕ and θ are the azimuth and elevation of the original recording. There are 8 such transformations, with the original recording being one of them. Consequently, we can increase our training set size from 4.2 to 33.6 hours.

3.4 Data preprocessing

To use audio recordings of varying durations in our recurrent model, several preprocessing steps are required. For the detection, we use each recording as its own data instance. For batching, we have to pad them to the same length and appropriately handle them when calling the GRU layers. In localization, this did not yield good results. To address this, we split the data into 5 second chunks to get a fixed length for all instances. This also makes the data suitable for non-recurrent models, such as the Conformer model.

As in the original implementation, we also experimented with standardising each of the $7 \cdot 64$ features. The original, augmented and normalized spectrograms were all computed in advance for faster loading.

3.5 Sound event detection and localization (SELD)

To extend our model to localization, we use the *activity-coupled Cartesian direction of arrival (ACCDOA)* approach from [Shi+21]. The idea is to predict the Cartesian coordinates instead of the azimuth and elevation angles. The activity of an event is represented by the vector length. If $a_{ct} \in \{0, 1\}$ is the reference activity of class c at time frame t, and $\mathbf{R}_{ct}$ is the direction of arrival unit vector, then the ACCDOA representation $\mathbf{P} \in \mathbb{R}^{3 \times C \times T}$ for this event is:

$$\mathbf{P}_{ct} = a_{ct}\mathbf{R}_{ct}.$$

To get the correct output, we add two linear layers to our backbone, one of size 128 and one of size $3 \cdot C$, followed by $\tanh$ activation. This gives us (x, y, z) coordinates for each class in each time frame. When predicting, an event is deemed active if the DOA vector length is larger than 0.5. The limitation of this approach is that we cannot detect simultaneous events of the same class.

In our data, we also have the distance information, which can be used to help the model. In [KPM24], they proposed a *multi-task* approach to learning. Instead of just the aforementioned DOA branch, we add another branch on top of our backbone for *sound distance estimation (SDE)*. It also contains two linear layers, but this time the latter one needs just C outputs followed by ReLU to model the distance.

Both branches are trained using MSE loss, with an additional weight parameter to balance their influences.

3.6 ResNet-Conformer

For the improved SELD model, we implemented the architecture proposed by the winning team in DCASE2022 [Wan+22]. First, we use 3 ResNet blocks (2 convolutions with kernels of size 3, both followed by batch normalisation, ReLU and dropout, with 2×1 average pooling and residual connection) with 24, 48 and 96 filters. A linear layer projects the filters to vectors of size 128, which are input to 8 Conformer blocks, as defined in the original paper [Gul+20] (feed-forward module, multi-head attention module, convolution module, feed-forward module). To get the correct output dimension, we also have to do time pooling, for which we use average pooling with kernel size 5. The DOA and SDE heads remain the same.

3.7 Evaluation metrics

For the evaluation, we used the metrics already implemented for this challenge [Pol+20b]. Evaluation is done on 1 second time segments for each class. Misses are counted for FN_c, while the Hungarian algorithm is used to obtain TP_c.

Model	Params	Normalize	Augment	Distance	$\mathbf{ER}_{20^\circ}$	$\mathbf{F}_{20^\circ}^{\text{micro}}$	$\mathbf{F}_{20^\circ}^{\text{macro}}$	LE	LR	@Epoch
SELDnet	750 k				0.76	0.29	0.17	46.2	0.33	189
		✓			0.76	0.31	0.18	52.3	0.33	221
			✓		0.62	0.44	0.21	48.7	0.30	37
				✓	0.79	0.26	0.11	93.3	0.24	246
Conformer	3.4 M				0.78	0.34	0.21	56.3	0.38	68
		✓	✓		**0.76**	0.44	0.26	45.7	0.51	39
			✓		**0.73**	**0.45**	**0.27**	**38.1**	**0.52**	39
			✓	✓	0.76	0.37	0.16	87.6	0.31	39

Table 1: Evaluation results for the baseline and the Conformer SELD systems. Each model is trained multiple times in different configurations (features standardisation, data augmentation, distance estimation). We use metrics from the DCASE Challenge, and indicate in which epoch the results were achieved.

To evaluate the localization further, an angle threshold of 20° is used to obtain $\text{FP}_{c,\geq 20^\circ}$ and $\text{TP}_{c,\leq 20^\circ}$.

The obtained counts are used to calculate the location-dependent *error rate* ($\text{ER}_{\leq 20^\circ}$) and *F1 score* ($\text{F}_{\text{c},\leq 20^\circ}$), with the latter being macro-averaged over all classes. Localization accuracy is also evaluated through *localization error* LE_c (mean angular error of matched true positives) and *localization recall* $\text{LR}_\text{c} = \text{TP}_c/(\text{TP}_c + \text{FN}_c)$, both of which get macro-averaged as well.

4 Results

All evaluations are done on the test split of the development set, which acts as our validation set. The final submissions to the competition were supposed to be done on a separate evaluation set, but as we already mentioned, we do not have its annotations, so we do not use it.

The SED models were trained for 200 epochs without data augmentation and for 20 with it. The results are shown in Table 2, and as we can see, the sigmoid loss gives better results overall. Using data augmentation improves the results just slightly when using the sigmoid loss function, but for softmax, the results are worse. For a larger increase, even more data would probably be needed because the model strongly overfits on the training set.

We can also notice that the micro-averaged F1 score is much higher than the macro average. This is caused by many classes with only a few annotations, which the model fails to detect. We tried using some sort of weighting of the loss function, but nothing helped with their recall. The final observation we make is that the compromise between the loss functions is the balance between precision and recall, since the model trained using the sigmoid loss has a higher recall, while the model trained using the softmax loss has a higher precision.

Loss	Augment	$\mathbf{F}_1^{\text{micro}}$	$\mathbf{F}_1^{\text{macro}}$	Pr	Re
Sigmoid		0.50	0.35	0.43	0.58
	✓	**0.53**	**0.36**	**0.49**	**0.59**
Softmax		0.50	0.31	0.56	0.46
	✓	0.50	0.30	0.51	0.48

Table 2: Evaluation results for the baseline SED system with two different loss functions. First, we trained models without and then with data augmentation. We use the macro-averaged F1 score as the main metric (as in SELD), but also include micro-averaged F1 score, precision (Pr) and recall (Re).

The results for the SELD systems are shown in Table 1. The models were trained for 250 epochs without data augmentation and for 40 with it. The models can overfit because of a smaller training set, so we also indicate the epoch in which the best results were obtained. For a proper challenge submission, this could be used as an early stopping measure. Over the whole training period, the train loss keeps dropping, but the validation loss quickly plateaus, yet the performance keeps improving slowly up to some point.

The data normalization does not improve the performance significantly. In the baseline model, the localization error even increases with it. The data augmentation in this case helps a lot more than in SED. For both models, distance estimation significantly worsens the performance, which could be caused by a lack of data, since distance estimation is more difficult than estimating DOA. When training the Conformer model using alternative configurations, we always use data augmentation because the model is a lot larger, so we do not want to train it using a very small dataset.

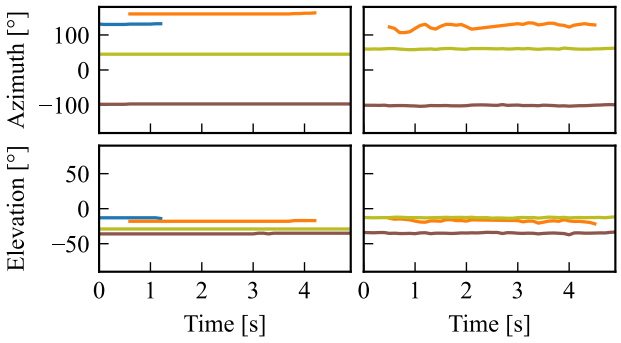

Figure 1: Labels (left) and predictions (right) of the azimuth and elevation angles for a 5 second segment from the test split containing 4 events of different classes. We used the Conformer model with standardised features and data augmentation.

As with SED, the macro-averaged F1 score is still a lot higher. The problem is again in the rare classes, for which

not many annotations are available. With the Conformer, the difference is smaller. The model is capable of detecting more events, which is also indicated by a higher localization recall.

Finally, we also show an example of predictions for the best model, see Figure 1. In this example, the model fails to detect one of the events, a limitation that frequently occurs with shorter or less frequent events. Another common issue arises when events are correctly detected but inaccurately localized. A further challenge, although less evident in this case, is the temporal imprecision in event detection. For instance, an event that is active for only one second may be erroneously predicted to span the entire five-second segment.

5 Discussion

In our implementation, we started with a simpler problem and showed that the baseline model is capable of detecting sound events in audio recordings. By choosing an appropriate loss function and using data augmentation, we managed to increase the model's performance.

The same model was then expanded to sound localization. We tried adding the distance estimation branch to our model but did not succeed in making it perform better than without it. The main problem with our implementations remains the lack of data. The baseline model was originally also trained using the synthetic data from 2021, while other teams mostly used some external datasets.

The Conformer is a large model, and accordingly, it requires a substantial amount of data. In the challenge, some teams used even larger variants, training them on over 100 hours of data, as well as additional data augmentations, further increasing the effective dataset size. In our case, all models tend to overfit, highlighting the need for more data. However, the core issue in this domain remains the limited availability of data and the high cost of acquiring it.

References

[Ada+19] Sharath Adavanne et al. "Sound Event Localization and Detection of Overlapping Sources Using Convolutional Recurrent Neural Networks". In: *IEEE Journal of Selected Topics in Signal Processing* 13.1 (Mar. 2019), pp. 34–48. ISSN: 1941-0484. DOI: 10.1109/jstsp.2018.2885636. URL: http://dx.doi.org/10.1109/JSTSP.2018.2885636.

[Art25] Daniel Arteaga. *Introduction to Ambisonics*. May 2025. DOI: 10.5281/zenodo.7963105.

[Gul+20] Anmol Gulati et al. *Conformer: Convolution-augmented Transformer for Speech Recognition*. 2020. arXiv: 2005.08100 [eess.AS]. URL: https://arxiv.org/abs/2005.08100.

[KPM24] Daniel Aleksander Krause, Archontis Politis, and Annamaria Mesaros. *Sound Event Detection and Localization with Distance Estimation*. 2024. arXiv: 2403.11827 [cs.SD]. URL: https://arxiv.org/abs/2403.11827.

[Mes+21] Annamaria Mesaros et al. "Sound Event Detection: A tutorial". In: *IEEE Signal Processing Magazine* 38.5 (Sept. 2021), pp. 67–83. ISSN: 1558-0792. DOI: 10.1109/msp.2021.3090678. URL: http://dx.doi.org/10.1109/MSP.2021.3090678.

[Pol+20a] Archontis Politis et al. "Overview and Evaluation of Sound Event Localization and Detection in DCASE 2019". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29 (2020), pp. 684–698. URL: https://ieeexplore.ieee.org/abstract/document/9306885.

[Pol+20b] Archontis Politis et al. "Overview and Evaluation of Sound Event Localization and Detection in DCASE 2019". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29 (2020), pp. 684–698. URL: https://ieeexplore.ieee.org/abstract/document/9306885.

[Pol+22] Archontis Politis et al. "STARSS22: A dataset of spatial recordings of real scenes with spatiotemporal annotations of sound events". In: *Proceedings of the 8th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*. Nancy, France, Nov. 2022, pp. 125–129. URL: https://dcase.community/workshop2022/proceedings.

[Pol+23] Archontis Politis et al. *STARSS23: Sony-TAu Realistic Spatial Soundscapes 2023*. Version 1.1.0. Zenodo, Mar. 2023. DOI: 10.5281/zenodo.7880637. URL: https://doi.org/10.5281/zenodo.7880637.

[Shi+21] Kazuki Shimada et al. *ACCDOA: Activity-Coupled Cartesian Direction of Arrival Representation for Sound Event Localization and Detection*. 2021. arXiv: 2010.15306 [eess.AS]. URL: https://arxiv.org/abs/2010.15306.

[Shi+23] Kazuki Shimada et al. "STARSS23: An Audio-Visual Dataset of Spatial Recordings of Real Scenes with Spatiotemporal Annotations of Sound Events". In: *In arXiv e-prints: 2306.09126* (2023). URL: https://arxiv.org/abs/2306.09126.

[Wan+22] Qing Wang et al. *THE NERC-SLIP SYSTEM FOR SOUND EVENT LOCALIZATION AND DETECTION OF DCASE2022 CHALLENGE*. Tech. rep. DCASE2022 Challenge, June 2022.

[Wan+23] Qing Wang et al. *A Four-Stage Data Augmentation Approach to ResNet-Conformer Based Acoustic Modeling for Sound Event Localization and Detection*. 2023. arXiv: 2101.02919 [cs.SD]. URL: https://arxiv.org/abs/2101.02919.

[Yas+20] Masahiro Yasuda et al. *Sound Event Localization based on Sound Intensity Vector Refined By DNN-Based Denoising and Source Separation*. 2020. arXiv: 2002.05994 [eess.AS]. URL: https://arxiv.org/abs/2002.05994.