ZERO- AND FEW-SHOT SOUND EVENT LOCALIZATION AND DETECTION

Kazuki Shimada[⋆†], *Kengo Uchida*[⋆], *Yuichiro Koyama*[‡], *Takashi Shibuya*[⋆], *Shusuke Takahashi*[‡], *Yuki Mitsufuji*[⋆‡], *Tatsuya Kawahara*[†]

[⋆] Sony AI [†] Kyoto University [‡] Sony Group Corporation

ABSTRACT

Sound event localization and detection (SELD) systems estimate direction-of-arrival (DOA) and temporal activation for sets of target classes. Neural network (NN)-based SELD systems have performed well in various sets of target classes, but they only output the DOA and temporal activation of preset classes trained before inference. To customize target classes after training, we tackle zero- and few-shot SELD tasks, in which we set new classes with a text sample or a few audio samples. While zero-shot sound classification tasks are achievable by embedding from contrastive language-audio pretraining (CLAP), zero-shot SELD tasks require assigning an activity and a DOA to each embedding, especially in overlapping cases. To tackle the assignment problem in overlapping cases, we propose an embed-ACCDOA model, which is trained to output track-wise CLAP embedding and corresponding activity-coupled Cartesian direction-of-arrival (ACCDOA). In our experimental evaluations on zero- and few-shot SELD tasks, the embed-ACCDOA model showed better location-dependent scores than a straightforward combination of the CLAP audio encoder and a DOA estimation model. Moreover, the proposed combination of the embed-ACCDOA model and CLAP audio encoder with zero- or few-shot samples performed comparably to an official baseline system trained with complete train data in an evaluation dataset.

***Index Terms*—** Sound event localization and detection (SELD), contrastive language-audio pretraining (CLAP)

1. INTRODUCTION

Given multichannel audio signals, a sound event localization and detection (SELD) system simultaneously estimates the direction-of-arrival (DOA) and temporal activation of target classes. SELD plays an essential role in many applications, such as surveillance [1], biodiversity monitoring [2], and smart devices [3, 4]. Each application has its own set of target sound event classes. For example, a surveillance SELD system is required to detect and localize screams, gunshots, or glass breaking, whereas a smart home SELD system needs to detect and localize speech, footsteps, or dog barking.

Recent neural network (NN)-based SELD systems usually set the target classes around ten sound events [5–14]. After training with annotated multichannel audio data, the NN-based systems detect and localize target sound events. There are two main approaches to associate a detection result with its DOA: class-wise and track-wise. An example of the class-wise approach is SELDnet [7], which outputs an activity and a DOA of each target class. An activity-coupled Cartesian DOA (ACCDOA) vector assigns the activity to the length of a Cartesian DOA vector [8], and the ACCDOA vector enables us to unify the activity and DOA branches into an ACCDOA branch. Event independent network v2 (EINV2) is a track-wise method, in which each track estimates an event's class and the corresponding location [9]. These SELD systems have shown a reasonable performance in both simulated and real environments [5–14].

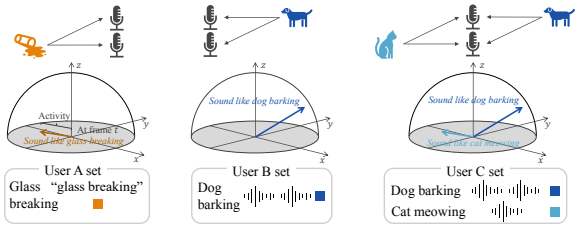

Fig. 1. Overview of zero- and few-shot SELD system.

However, they only output the temporal activation and DOA of preset classes trained before inference. This is problematic because users generally prefer their own set of target sound event classes. To customize target classes after training, we tackle zero- and few-shot SELD tasks, in which we set new classes with zero- and few-shot samples (Fig. 1). In this context, a zero-shot sample means a text sample of sound events, e.g., "glass breaking," and few-shot samples mean a few audio samples of sound events.

Contrastive language-audio pretraining (CLAP) allows for zero-shot tasks [15–17], similar to contrastive language-image pretraining (CLIP) [18, 19]. CLAP learns to connect language and audio using two encoders and contrastive learning to bring audio and text descriptions into a joint multi-modal space. Zero-shot classification is solved by computing the cosine similarity between the CLAP embeddings of an audio query and text support samples [15–17]. When we have a few audio samples of target classes, we can tackle few-shot audio tasks. In addition to the few-shot classification task [20], several works have tackled the few-shot sound event detection (SED) task [21, 22]. A few-shot SED system takes an audio query sequence and needs to estimate sections without target classes, i.e., background noise sections [22]. While the CLAP embeddings allow us to tackle zero- and few-shot classification and SED tasks, zero- and few-shot SELD tasks have other requirements: specifically, they must output an event's embedding and its corresponding DOA in both single source and overlapping cases, as shown in Fig. 2.

In this paper, to solve the assignment problem in overlapping cases, we propose an embed-ACCDOA model, which is trained to output track-wise CLAP embedding and the corresponding ACCDOA vector. The embed-ACCDOA model uses a similar network architecture to the track-wise SELD method EINV2 [9]. Before inference, we obtain support embeddings from zero- or few-shot samples of target classes. In the inference, if activity from the estimated ACCDOA vector is larger than a threshold, the system outputs DOA from the ACCDOA vector and a class whose support embedding is the nearest to the estimated embedding. We also propose combining the embed-ACCDOA model and the CLAP audio encoder to utilize a CLAP embedding itself in single-source cases. To investigate the proposed methods in zero- and few-shot SELD tasks, we first train

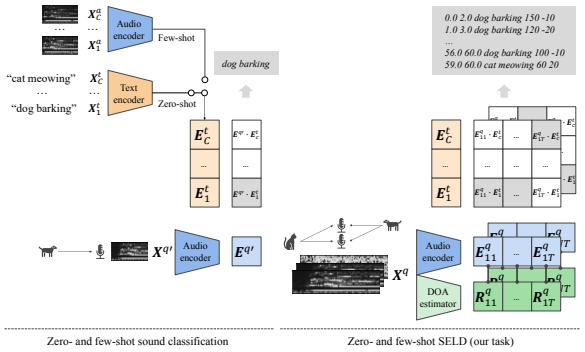

Fig. 2. Zero- and few-shot sound classification and SELD tasks.

the embed-ACCDOA model with a synthetic dataset, and then evaluate the system with two datasets: Sony-TAU Realistic Spatial Soundscapes 2023 (STARSS23) [6, 23] and TAU-NIGENS Spatial Sound Events 2021 (TNSSE21) [5]. We prepare a straightforward combination of the CLAP audio encoder and a DOA estimation (DOAE) model for comparison. We also compare with the official baseline SELD systems trained with full training datasets for reference.

2. ZERO- AND FEW-SHOT SELD TASKS

We first explain zero-shot sound classification tasks using language-audio models such as CLAP [15–17]. The left side of Fig. 2 shows a zero-shot sound classification task. We define a zero-shot support sample of class c as a text data $\boldsymbol{X}_c^t$, e.g., "`class-name`." Let an audio spectrogram $\boldsymbol{X}^{q'} \in \mathbb{R}^{F' \times T'}$ be a query sample. F' and T' indicate the numbers of frequency bins and time frames, respectively. The D-dimensional embedding of text support $\boldsymbol{E}_c^t \in \mathbb{R}^D$ and one of audio query $\boldsymbol{E}^{q'} \in \mathbb{R}^D$ are respectively obtained by a CLAP text encoder $\mathcal{F}_{\mathrm{CLAPtext}}$ and a CLAP audio encoder $\mathcal{F}_{\mathrm{CLAPaudio}}$:

$$\boldsymbol{E}_c^t = \mathcal{F}_{\mathrm{CLAPtext}}(\boldsymbol{X}_c^t), \tag{1}$$

$$\boldsymbol{E}^{q'} = \mathcal{F}_{\mathrm{CLAPaudio}}(\boldsymbol{X}^{q'}). \tag{2}$$

For C classes, we construct C prompt texts $\boldsymbol{X}^t = \{\boldsymbol{X}_c^t\}_{c=1,\dots,C}$. For a given audio $\boldsymbol{X}^{q'}$, we determine the best match $\boldsymbol{X}_c^t$ among $\boldsymbol{X}^t$ by the cosine similarity function over their embeddings $\boldsymbol{E}^{q'}$ and $\boldsymbol{E}_c^t$.

We keep the same support embeddings $\boldsymbol{E}_c^t$ for the zero-shot SELD tasks. Unlike zero-shot sound classification tasks, zero-shot SELD tasks need to output per time frame, to consider overlapping sound events, and to associate embeddings and DOAs. The right side of Fig. 2 depicts a zero-shot SELD task, given a multichannel audio query sequence $\boldsymbol{X}^q \in \mathbb{R}^{M \times F \times T}$, where M, F, and T indicate the numbers of feature channels, frequency bins, and time frames, respectively. To solve the requirements for time frames and overlapping sound events, the audio query embeddings should have dimensions of time and track, i.e., $\boldsymbol{E}^q \in \mathbb{R}^{D \times N \times T}$, where N indicates the numbers of output tracks. Also, audio query embeddings $\boldsymbol{E}^q$ need to be associated with Cartesian DOA vectors $\boldsymbol{R}^q \in \mathbb{R}^{3 \times N \times T}$. For each audio query embedding $\boldsymbol{E}_{nt}^q$, we determine the best match $\boldsymbol{E}_c^t$ among $\boldsymbol{E}^t = \{\boldsymbol{E}_c^t\}_{c=1,\dots,C}$ by the cosine similarity function.

When we use a K-shot audio support set $\boldsymbol{X}_c^a = \{\boldsymbol{X}_{ck}^a \in \mathbb{R}^{F' \times T'}\}_{k=1,\dots,K}$ instead of a zero-shot text sample for a class c, we replace the text embedding $\boldsymbol{E}_c^t$ with an audio embedding $\boldsymbol{E}_c^a \in \mathbb{R}^D$:

$$\boldsymbol{E}_c^a = \frac{1}{K} \sum_{\boldsymbol{X}_{ck}^a \in \boldsymbol{X}_c^a} \mathcal{F}_{\mathrm{CLAPaudio}}(\boldsymbol{X}_{ck}^a), \tag{3}$$

where we use an average of the embeddings called a prototype [24].

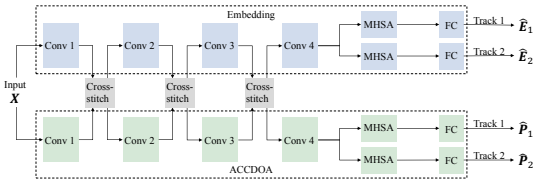

Fig. 3. Overview of a 2-track embed-ACCDOA model.

3. METHOD

3.1. Embed-ACCDOA model

To achieve zero-shot SELD tasks, a CLAP model can output embeddings in single source cases. However, CLAP models are not designed to output each embedding of sound events in overlapping cases. To obtain embeddings and the corresponding DOAs in overlapping cases, we propose an embed-ACCDOA model, which outputs an embedding and an ACCDOA vector in each track. The model is trained to output an oracle CLAP embedding and its corresponding ACCDOA vector in a track, given a multichannel spectrogram $\boldsymbol{X} \in \mathbb{R}^{M \times F \times T}$. The embed-ACCDOA format is formulated by embeddings, $\boldsymbol{E} \in \mathbb{R}^{D \times N \times T}$, and ACCDOA vectors, $\boldsymbol{P} \in \mathbb{R}^{3 \times N \times T}$.

Each ACCDOA vector of a track is represented by three nodes corresponding to the sound event location in the x, y, and z axes [8]. Let $\boldsymbol{a} \in \mathbb{R}^{N \times T}$ be activities, whose reference value is $a_{nt}^* \in \{0, 1\}$, i.e., it is 1 when the event is active and 0 when inactive. Also, let $\boldsymbol{R} \in \mathbb{R}^{3 \times N \times T}$ be Cartesian DOAs, where the length of each Cartesian DOA is 1, i.e., $||\boldsymbol{R}_{nt}|| = 1$ when a track n is active. $||\cdot||$ is the L2 norm. An ACCDOA vector is formulated as follows [8]:

$$\boldsymbol{P}_{nt} = a_{nt} \boldsymbol{R}_{nt}. \tag{4}$$

An activity and a Cartesian DOA vector are obtained from the ACCDOA vector [8]:

$$a_{nt} = ||\boldsymbol{P}_{nt}||, \tag{5}$$

$$\boldsymbol{R}_{nt} = \frac{\boldsymbol{P}_{nt}}{||\boldsymbol{P}_{nt}||}. \tag{6}$$

In training, we use synthetic mixture with J clean directional sound events $\{\boldsymbol{X}_j \in \mathbb{R}^{M \times F \times T}\}_{j=1,\dots,J}$:

$$\boldsymbol{X} = \sum_j^J \boldsymbol{X}_j + \boldsymbol{N}, \tag{7}$$

where $\boldsymbol{N} \in \mathbb{R}^{M \times F \times T}$ is ambient noise. To obtain an oracle embedding $\boldsymbol{E}_{jt}^*$, we use CLAP audio embeddings from clean events:

$$\boldsymbol{E}_{jt}^* = \mathcal{F}_{\mathrm{CLAPaudio}}(\boldsymbol{X}_j^{(1)}), \tag{8}$$

where we utilize the entire length event of the first channel $\boldsymbol{X}_j^{(1)}$ since one frame of the spectrogram is too short to obtain an accurate CLAP embedding. If the number of events is less than the number of tracks, we set a zero vector as an oracle embedding.

To output embeddings and ACCDOA vectors in a track-wise manner, the embed-ACCDOA model uses a similar network architecture to EINV2 [9], a well-known track-wise model. Our architecture has two branches: an embedding branch and an ACCDOA branch. Each branch consists of convolution blocks, multi-head self-attention (MHSA) blocks, and fully connected layers. Also, the convolution blocks use cross-stitch units [9, 25] to share parameters between the two branches. The architecture is depicted in Fig. 3.

Similarly to other track-wise approaches, the embed-ACCDOA model also suffers from the track permutation problem. To overcome this issue, we adopt permutation-invariant training (PIT) for the training process. The frame-level PIT [9] is used in this study. Assume all possible permutations constitute a permutation set Perm. $\alpha \in \mathrm{Perm}(t)$ is one possible frame-level permutation at frame t. A PIT loss for the embed-ACCDOA format can be written as follows:

$$\mathcal{L}^{\mathrm{PIT}} = \frac{1}{T}\sum_{t}^{T} \min_{\alpha \in \mathrm{Perm}(t)} l^{\mathrm{EA}}_{\alpha,t}, \tag{9}$$

$$l^{\mathrm{EA}}_{\alpha,t} = \frac{1}{N}\sum_{n}^{N} \beta_E l^{\mathrm{E}}_{\alpha,nt} + \beta_A l^{\mathrm{A}}_{\alpha,nt}, \tag{10}$$

$$l^{\mathrm{E}}_{\alpha,nt} = \mathrm{CosineSimilarity}(\boldsymbol{E}^{*}_{\alpha,nt}, \hat{\boldsymbol{E}}_{nt}), \tag{11}$$

$$l^{\mathrm{A}}_{\alpha,nt} = \mathrm{MSE}(\boldsymbol{P}^{*}_{\alpha,nt}, \hat{\boldsymbol{P}}_{nt}), \tag{12}$$

where $\boldsymbol{E}^{*}_{\alpha,nt}$ and $\boldsymbol{P}^{*}_{\alpha,nt}$ are respectively an oracle embedding and ACCDOA vector of a permutation α, and $\hat{\boldsymbol{E}}_{nt}$ and $\hat{\boldsymbol{P}}_{nt}$ are respectively a predicted embedding and ACCDOA vector, at track n and frame t. β_E and β_A are loss coefficients for embedding and ACCDOA, respectively. We use cosine similarity as a loss function for embeddings, as it is widely used in zero-shot tasks. We use mean squared error (MSE) as a loss function for the ACCDOA vectors [8].

In inference, we prepare each support embedding $\boldsymbol{E}^{s}_{c} \in \{\boldsymbol{E}^{t}_{c}, \boldsymbol{E}^{a}_{c}\}$ as described in Section 2. Given a multichannel query sequence $\boldsymbol{X}^{q}$, the embed-ACCDOA model estimates embedding $\hat{\boldsymbol{E}}$ and the corresponding ACCDOA $\hat{\boldsymbol{P}}$. Class at track n and frame t, $\hat{c}_{nt}$, is obtained from the embedding as follows:

$$\hat{c}_{nt} = \underset{c \in C}{\mathrm{argmax}}\, \mathrm{CosineSimilarity}(\hat{\boldsymbol{E}}_{nt}, \boldsymbol{E}^{s}_{c}). \tag{13}$$

To obtain the final outputs of class and DOA at frame t, $(\hat{c}_t, \hat{\boldsymbol{R}}_t)^l$, where $l \in \{0, 1, ..., n\}$, we use two thresholds: σ_a for the track with the highest activity, and $\sigma_b (> \sigma_a)$ for other tracks.

$$\mathrm{Threshold}_{nt} = \begin{cases} \sigma_a & \text{if } n = \mathrm{argmax}_{n' \in N}\, a_{n't}, \\ \sigma_b & \text{if } n \neq \mathrm{argmax}_{n' \in N}\, a_{n't}. \end{cases} \tag{14}$$

Since single source cases are easier to estimate than overlapping cases, a lower σ_a can increase true positives while a higher σ_b can prevent false positives.

We also incorporate a support embedding for background noise $\boldsymbol{E}^{s}_{\mathrm{noise}}$ to decrease false positives. If the support embedding for noise is more similar to an estimated embedding than the target classes, we set no event at the frame. In a zero-shot setting, we obtain the support embedding with the text data of "silent." We take a few audio samples without target classes in a few-shot setting.

3.2. Combination of CLAP and Embed-ACCDOA

To improve the zero-shot SELD performance in single source cases, we combine the embed-ACCDOA model and the CLAP audio encoder in inference. When there is only one source after the thresholding, we calculate the cosine similarity between the support embeddings and the embedding from the CLAP encoder $\hat{\boldsymbol{E}}_{t,\mathrm{CLAP}}$ instead of the embedding predicted by embed-ACCDOA. We finally use the class $\hat{c}_{t,\mathrm{CLAP}}$, which is calculated as

$$\hat{\boldsymbol{E}}_{t,\mathrm{CLAP}} = \mathcal{F}_{\mathrm{CLAPaudio}}(\boldsymbol{X}^{q,(1)}), \tag{15}$$

$$\hat{c}_{t,\mathrm{CLAP}} = \underset{c \in C}{\mathrm{argmax}}\, \mathrm{CosineSimilarity}(\hat{\boldsymbol{E}}_{t,\mathrm{CLAP}}, \boldsymbol{E}^{s}_{c}), \tag{16}$$

where we utilize the entire length signal of the first channel $\boldsymbol{X}^{q,(1)}$, following the training phase.

4. EXPERIMENTAL EVALUATIONS

We evaluate the embed-ACCDOA methods in zero- and few-shot SELD tasks using multichannel audio data with the first-order Ambisonics (FOA) format. We compare the proposed methods with a straightforward combination of a CLAP audio encoder and a DOAE model. The proposed methods are also compared with the official baseline SELD systems trained with full training datasets.

4.1. Task setups

To set up the zero- and few-shot SELD tasks, we prepare training data using a data generator[1] from the TAU Spatial Room Impulse Response Database (TAU-SRIR DB)[2] and Freesound Dataset 50k (FSD50K) [26]. The data generator and the data of spatial room impulse response (SRIR) and noise are used to synthesize a part of the training data for DCASE2023 Challenge Task 3. While the synthetic data for the challenges chose FSD50K samples to match the target classes, our training data for zero- and few-shot SELD tasks used all the FSD50K training samples. Finally, 2,250 one-minute spatial mixtures are synthesized using the measured SRIRs and noise from nine rooms and the samples from FSD50K.

As an evaluation dataset for the zero- and few-shot SELD tasks, we use the development set of STARSS23 [6, 23]. The recordings contain 13 target sound event classes such as footsteps and bell. The development set of STARSS23 totals about 7 hours and 22 minutes, of which 168 clips are recorded with 57 participants in 16 rooms. The development set is further split into `dev-set-train` (90 clips) and `dev-set-test` (78 clips). K-shot audio samples of the 13 target classes are extracted from `dev-set-train` in the few-shot setting, while the zero-shot setting does not use audio samples. We use `dev-set-test` as query sequences in the evaluation.

The proposed methods can use other sets of target classes without re-training. To check the performance in another dataset, we prepare the development set of TNSSE21 [5] as an additional evaluation dataset. The data are synthesized by adding sound event samples convolved with SRIR to spatial ambient noise. The SRIRs and ambient noise recordings are collected at 15 different indoor locations. The sound event samples consist of 12 event classes, such as crying baby and barking dog. The dataset contains 600 one-minute sound scene recordings: 400 for training, 100 for validation, and 100 for testing. K-shot audio samples are extracted from the training split in the few-shot setting. We omit the validation split and use the test split as query sequences.

Following the setup, five metrics are used for the evaluation [27]. The first two metrics are the location-dependent error rate ER_{20° and F-score F_{20°, where predictions are considered true positives only when the distance from the reference is less than 20°. The next is the localization error LE_{CD}, which expresses the average angular distance between the same class's predictions and references. The fourth is a simple localization recall metric LR_{CD}, which tells the true positive rate of how many of these localization estimates are detected in a class out of the total number of class instances. We also adopt an aggregated SELD error, $\mathcal{E}_{\mathrm{SELD}}$, which is defined as

$$\mathcal{E}_{\mathrm{SELD}} = \frac{\mathrm{ER}_{20^\circ} + (1 - \mathrm{F}_{20^\circ}) + \frac{\mathrm{LE}_{\mathrm{CD}}}{180^\circ} + (1 - \mathrm{LR}_{\mathrm{CD}})}{4}. \tag{17}$$

[1] https://github.com/danielkrause/DCASE2022-data-generator

[2] https://zenodo.org/record/6408611

Table 1. SELD performance of zero- and few-shot methods evaluated for the test split of the STARSS23 development set.

Method	# of shots	ER_{20°	F_{20°	LE_{CD}	LR_{CD}	$\mathcal{E}_{SELD}$
Combination of CLAP and DOAE	Zero	0.860	11.2	38.4	40.8	0.638
	Few 10	0.837	14.3	36.2	46.5	0.607
Embed-ACCDOA (proposed)	Zero	0.835	15.8	55.2	29.9	0.671
	Few 10	0.777	19.3	27.0	34.1	0.598
Combination of CLAP and Embed-ACCDOA (proposed)	Zero	0.773	18.7	51.9	36.1	0.628
	Few 10	0.756	19.2	35.0	40.2	0.589
Official baseline trained with full training dataset	(Full)	(0.594)	(29.4)	(23.4)	(49.8)	(0.483)

4.2. Experimental settings

The embed-ACCDOA model uses the EINV2 network architecture [9] with slight modification. The difference between the original and the one used here is the output size of the final fully connected layer in the embedding branch, i.e., from the number of classes to the embedding size. The embedding size is 512, following a previous CLAP implementation [16]. The number of tracks in the embed-ACCDOA format is fixed at 3. The loss coefficients for embedding and ACCDOA are set to 0.6 and 0.4, respectively. We set the threshold for the track with the highest activity to 0.2. We also set the threshold for the other tracks to 0.8.

We compare the embed-ACCDOA methods with a straightforward combination of a CLAP audio encoder [16] and a DOAE model with a one-track one-class ACCDOA format. Since the DOAE model outputs only one ACCDOA vector per time frame, the network architecture of the model is set equivalent to the ACCDOA branch of the one-track embed-ACCDOA model. The training data is the same. The loss function is MSE between the oracle and the estimated ACCDOA vectors. In inference, we simply assign the ACCDOA vector output to the class output from the CLAP audio encoder at each frame. We set the threshold to 0.2.

Other configurations mostly follow the multi-ACCDOA paper [28] in all methods. Multichannel amplitude spectrograms and inter-channel phase differences (IPDs) are used as features. Two data augmentation methods are applied: equalized mixture data augmentation (EMDA) [29] and rotation in FOA [30]. The sampling frequency is set to 24 kHz. The short-term Fourier transform (STFT) is applied with a 20-ms frame length and a 10-ms frame hop. Input features are segmented to have a fixed length of 1.27 seconds. The shift length is set to 1.2 seconds during inference. We use a batch size of 32, and each training sample is generated on the fly. We use the Adam optimizer with a weight decay of 10^{-6}. We gradually increase the learning rate to 0.001 with 25,000 iterations [31]. After the warm-up, the learning rate is decreased by 10% if the SELD error of the validation did not improve in 20,000 consecutive iterations. We validate and save model weights every 5,000 iterations up to 200,000. Finally, we apply stochastic weight averaging (SWA) [32] to the last ten models.

We also run the official baseline systems for reference [5]. Note that the baseline systems are trained with full training datasets, while our methods take only zero- or few-shot samples of the target classes.

4.3. Experimental results

Table 1 summarizes the performance of the zero- and few-shot methods in the development set of STARSS23. The embed-ACCDOA model shows a similar SELD error to the combination of the CLAP audio encoder and the DOAE model. While the combination of DOAE and CLAP performs better in localization recall, the embed-ACCDOA model performs better in the location-dependent error rate and F-score. The proposed combination of the CLAP audio encoder and the embed-ACCDOA model achieves the best SELD error among the zero- and few-shot methods. The proposed combination tackles overlapping cases while borrowing the detection performance of the CLAP audio encoder in single-source cases. While gaps exist between the official baseline system trained with complete train data and the zero- and few-shot systems, the proposed methods show promising results without re-training on the target classes.

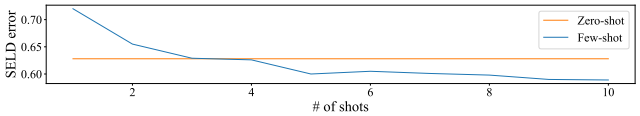

Fig. 4. SELD performance of the combination of the CLAP audio encoder and embed-ACCDOA model for STARSS23 with different numbers of shots.

Table 2. SELD performance of zero- and few-shot methods evaluated for the test split of the TNSSE21 development set.

Method	# of shots	ER_{20°	F_{20°	LE_{CD}	LR_{CD}	$\mathcal{E}_{SELD}$
CLAP and Embed-ACCDOA	Zero	1.008	25.8	26.2	46.9	0.607
	Few 10	1.009	24.9	31.0	51.5	0.604
Official baseline	(Full)	(0.706)	(26.0)	(37.0)	(40.4)	(0.562)

Fig. 4 shows the performance of the combination of CLAP and embed-ACCDOA with different numbers of shots. When the number increases, the method improves performance. The zero-shot performance is better than the 1-shot and comparable to the 3-shot.

Table 2 lists the performance of the zero- and few-shot SELD methods in the development set of TNSSE21. We use the same model as the experiments on STARSS23 to check the capability of adapting to another dataset with zero- and few-shot samples. The proposed method without re-training achieves comparable results to the official baseline system trained with complete train data.

5. CONCLUSION

We investigate zero- and few-shot sound event localization and detection (SELD) tasks, which enable us to customize the target classes of SELD systems with only a text sample or a few audio samples. While zero-shot sound classification tasks are achieved by embeddings from contrastive language-audio pretraining (CLAP) models, zero-shot SELD tasks require the assignment of an activity and a direction-of-arrival (DOA) to each embedding, especially in overlapping cases. To address the assignment problem in overlapping cases, we propose an embed-ACCDOA model, which is trained to output track-wise CLAP embedding and associated activity-coupled Cartesian DOA (ACCDOA). In our experimental evaluations on the zero- and few-shot SELD tasks, the embed-ACCDOA model shows a better location-dependent error rate and F-score than the straightforward combination of the CLAP audio encoder and the DOA estimation model. Moreover, the proposed combination of the embed-ACCDOA model and the CLAP audio encoder with zero- or few-shot samples shows comparable performance to the official baseline system trained with complete train data in an evaluation dataset. We will conduct comprehensive experiments in the future.

6. REFERENCES

[1] M. Crocco, M. Cristani, A. Trucco, and V. Murino, "Audio surveillance: A systematic review," *ACM Computing Surveys*, vol. 48, no. 4, pp. 1–46, 2016.

[2] S. Chu, S. Narayanan, and C.-C. J. Kuo, "Environmental sound recognition with time–frequency audio features," *IEEE Trans. on ASLP*, vol. 17, no. 6, pp. 1142–1158, 2009.

[3] N. Yalta, K. Nakadai, and T. Ogata, "Sound source localization using deep learning models," *Journal of Robotics and Mechatronics*, vol. 29, no. 1, pp. 37–48, 2017.

[4] H. Sun, X. Liu, K. Xu, J. Miao, and Q. Luo, "Emergency vehicles audio detection and localization in autonomous driving," *arXiv*, 2021.

[5] A. Politis, S. Adavanne, D. Krause, A. Deleforge, P. Srivastava, and T. Virtanen, "A dataset of dynamic reverberant sound scenes with directional interferers for sound event localization and detection," in *Proc. of DCASE Workshop*, 2021.

[6] 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," in *Proc. of DCASE Workshop*, 2022.

[7] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound event localization and detection of overlapping sources using convolutional recurrent neural networks," *IEEE JSTSP*, vol. 13, no. 1, pp. 34–48, 2018.

[8] K. Shimada, Y. Koyama, N. Takahashi, S. Takahashi, and Y. Mitsufuji, "ACCDOA: Activity-coupled cartesian direction of arrival representation for sound event localization and detection," in *Proc. of IEEE ICASSP*, 2021.

[9] 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. of IEEE ICASSP*, 2021.

[10] O. Slizovskaia, G. Wichern, Z.-Q. Wang, and J. Le Roux, "Locate this, not that: Class-conditioned sound event doa estimation," in *Proc. of IEEE ICASSP*, 2022.

[11] J. Hu, Y. Cao, M. Wu, F. Yang, Z. Yu, W. Wang, M. D. Plumbley, and J. Yang, "META-SELD: Meta-learning for fast adaptation to the new environment in sound event localization and detection," in *Proc. of DCASE Workshop*, 2023.

[12] J. S. Kim, H. J. Park, W. Shin, and S. W. Han, "AD-YOLO: You look only once in training multiple sound event localization and detection," in *Proc. of IEEE ICASSP*, 2023.

[13] Q. Wang, J. Du, Z. Nian, S. Niu, L. Chai, H. Wu, J. Pan, and C.-H. Lee, "Loss function design for DNN-based sound event localization and detection on low-resource realistic data," in *Proc. of IEEE ICASSP*, 2023.

[14] S. Niu, J. Du, Q. Wang, L. Chai, H. Wu, Z. Nian, L. Sun, Y. Fang, J. Pan, and C.-H. Lee, "An experimental study on sound event localization and detection under realistic testing conditions," in *Proc. of IEEE ICASSP*, 2023.

[15] B. Elizalde, S. Deshmukh, M. Al Ismail, and H. Wang, "CLAP: Learning audio concepts from natural language supervision," in *Proc. of IEEE ICASSP*, 2023.

[16] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *Proc. of IEEE ICASSP*, 2023.

[17] S. S. Kushwaha and M. Fuentes, "A multimodal prototypical approach for unsupervised sound classification," *arXiv*, 2023.

[18] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. of ICML*, 2021.

[19] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui, "Open-vocabulary object detection via vision and language knowledge distillation," in *Proc. of ICLR*, 2022.

[20] S.-Y. Chou, K.-H. Cheng, J.-S. R. Jang, and Y.-H. Yang, "Learning to match transient sound events using attentional similarity for few-shot sound recognition," in *Proc. of IEEE ICASSP*, 2019.

[21] Y. Wang, J. Salamon, N. J. Bryan, and J. P. Bello, "Few-shot sound event detection," in *Proc. of IEEE ICASSP*, 2020.

[22] K. Shimada, Y. Koyama, and A. Inoue, "Metric learning with background noise class for few-shot detection of rare sound events," in *Proc. of IEEE ICASSP*, 2020.

[23] K. Shimada, A. Politis, P. Sudarsanam, D. Krause, K. Uchida, S. Adavanne, A. Hakala, Y. Koyama, N. Takahashi, S. Takahashi *et al.*, "STARSS23: An audio-visual dataset of spatial recordings of real scenes with spatiotemporal annotations of sound events," in *Proc. of NeurIPS*, 2023.

[24] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. of NeurIPS*, 2017.

[25] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *Proc. of IEEE CVPR*, 2016.

[26] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50k: An open dataset of human-labeled sound events," *IEEE/ACM Trans. on ASLP*, vol. 30, pp. 829–852, 2021.

[27] A. Mesaros, S. Adavanne, A. Politis, T. Heittola, and T. Virtanen, "Joint measurement of localization and detection of sound events," in *Proc. of IEEE WASPAA*, 2019.

[28] 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. of IEEE ICASSP*, 2022.

[29] N. Takahashi, M. Gygli, and L. Van Gool, "AENet: Learning deep audio features for video analysis," *IEEE Trans. on Multimedia*, vol. 20, pp. 513–524, 2017.

[30] L. Mazzon, Y. Koizumi, M. Yasuda, and N. Harada, "First order ambisonics domain spatial augmentation for DNN-based direction of arrival estimation," in *Proc. of DCASE Workshop*, 2019.

[31] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch SGD: Training ImageNet in 1 hour," *arXiv*, 2017.

[32] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," in *Proc. of UAI*, 2018.