META-SELD: META-LEARNING FOR FAST ADAPTATION TO THE NEW ENVIRONMENT IN SOUND EVENT LOCALIZATION AND DETECTION

Jinbo Hu[1,2]*, Yin Cao*[3]*, Ming Wu*[1]*, Feiran Yang*[1]*, Ziying Yu*[1]*, Wenwu Wang*[4]*, Mark D. Plumbley*[4]*, Jun Yang*[1,2]

[1]Key Laboratory of Noise and Vibration Research, Institute of Acoustics, Chinese Academy of Sciences, Beijing, China, {hujinbo, mingwu, feiran, yuziying, jyang}@mail.ioa.ac.cn
[2]University of Chinese Academy of Sciences, Beijing, China
[3]Department of Intelligent Science, Xi'an Jiaotong Liverpool University, China, yin.k.cao@gmail.com
[4]Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, UK, {w.wang, m.plumbley}@surrey.ac.uk

ABSTRACT

For learning-based sound event localization and detection (SELD) methods, different acoustic environments in the training and test sets may result in large performance differences in the validation and evaluation stages. Different environments, such as different sizes of rooms, different reverberation times, and different background noise, may be reasons for a learning-based system to fail. On the other hand, acquiring annotated spatial sound event samples, which include onset and offset time stamps, class types of sound events, and direction-of-arrival (DOA) of sound sources is very expensive. In addition, deploying a SELD system in a new environment often poses challenges due to time-consuming training and fine-tuning processes. To address these issues, we propose Meta-SELD, which applies meta-learning methods to achieve fast adaptation to new environments. More specifically, based on Model Agnostic Meta-Learning (MAML), the proposed Meta-SELD aims to find good meta-initialized parameters to adapt to new environments with only a small number of samples and parameter updating iterations. We can then quickly adapt the meta-trained SELD model to unseen environments. Our experiments compare fine-tuning methods from pre-trained SELD models with our Meta-SELD on the Sony-TAU Realistic Spatial Soundscapes 2023 (STARSSS23) dataset. The evaluation results demonstrate the effectiveness of Meta-SELD when adapting to new environments.

***Index Terms*—** SELD, MAML, unseen environments, fast adaptation, meta-learning, few-shot

1. INTRODUCTION

Sound event localization and detection (SELD) refers to detecting categories, presence, and spatial locations of different sound sources. SELD characterizes sound sources in a spatial-temporal manner. SELD was first introduced in Task 3 of the Detection and Classification of Acoustics Scenes and Events (DCASE) 2019 Challenge [1]. After three iterations of Task 3 of the DCASE Challenge, types of data transform from computationally generated spatial recordings to real-scene recordings [2].

SELD can be regarded as a Multi-Task Learning problem. Adavanne et al. [3] proposed SELDnet for a joint task of sound event detection (SED) and regression-based direction-of-arrival (DOA) estimation. SELDnet is unable to detect homogeneous overlap, which refers to overlapping sound events of the same type but with different locations. The Event-Independent Network V2 (EINV2), with a track-wise output format and permutation invariant training, was proposed to tackle the homogeneous overlap detection problem [4–6]. Different from two outputs of SED and DOA in SELDnet and EINV2, the Activity-coupled Cartesian DOA (ACCDOA) approach merges two subtasks into a single task [7, 8]. The Cartesian DOA vectors contain the activity information of sound events in the ACCDOA method.

In practical SELD system deployment, unseen complex environments may lead to performance degradation. In the STARSS22 dataset [2], there are no duplicated recording environments in the training and validation sets. Our previous system submitted to Task 3 of the DCASE 2022 Challenge obtained the second rank in the team ranking [9]. However, we found unsatisfactory generalization performance for fold4_room2 recordings in the *dev-test-tau* set of STARSS22 [9]. Experimental results show that class-dependent localization error $\mathrm{LE_{CD}}$ is high and location-dependent F-score $\mathrm{F_{\leq 20^\circ}}$ is low, but class-dependent localization recall $\mathrm{LR_{CD}}$ is high. This suggests there may be the weak localizing performance of our system in fold4_room2. In addition, manually annotated spatial sound event recordings are very expensive. Taking the STARSS22 dataset for example [2], each scene was captured with a 32-channel spherical microphone array, a 360° camera, a motion capture (mocap) system, and wireless microphones. Onset, offset, and class information of sound events were manually detected and classified by annotators through listening to wireless microphone recordings and watching video recordings, while positional annotations were extracted for each event by masking the tracker data with the temporal activity window of the event. In the end, 360° video recordings are utilized to validate those annotations. This type of complex recording and annotation process means that large datasets of the annotated spatial recording might be expensive.

Few-shot learning can act as a test bed for learning like humans, allowing a system to learn from small samples and reducing data gathering effort and computation [10]. Meta-learning, which facilitates few-shot learning, learns a general-purpose learning algorithm that generalizes across tasks and ideally enables each new task to be learned well from the task-distribution view [11]. Meta-learning has advanced few-shot learning significantly in computer

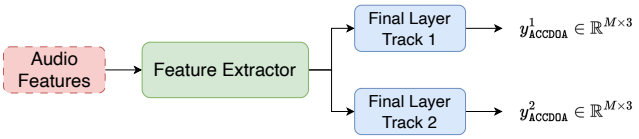

Figure 1: The multi-ACCDOA representation of the SELD model. There is no track dimension in the ACCDOA representation.

vision [12,13]. One of the most successful meta-learning algorithms is model-agnostic meta-learning (MAML) [14]. MAML tries to learn general initial parameters that can be rapidly adapted to another task. The method is model-agnostic and compatible with any model trained with gradient descent. It can be applicable to a variety of different learning problems, including classification, regression, and reinforcement learning. In audio signal processing, the meta-learning method has recently attracted interest as a way to solve few-shot learning problems recently. Meta-TTS [15] is proposed to build personalized speech synthesis systems with few enrolled recordings of unseen users' voices using MAML. In [16], MAML is utilized to allow sound source localization models to adapt to different environments and conditions.

In this paper, we propose Meta-SELD, applying meta-learning to SELD models with activity-coupled Cartesian DOA (ACCDOA) representation [7] to improve performance, especially in localization. We use MAML to find general initial parameters to minimize the loss across several tasks in Meta-SELD so that it can quickly adapt to an unseen environment. We take recordings in different environments as different tasks and aim to improve the performance of a specific unseen environment with a few samples recorded in the same environment. The experimental results demonstrate that Meta-SELD outperforms the fine-tuning method from the pre-trained SELD model in the STARSS23 dataset.

2. RELATED WORK

Activity-coupled Cartesian DOA (ACCDOA) representation [7] assigns a sound event activity to the length of a corresponding Cartesian DOA. When inferring, the threshold is set for the length of class-wise Cartesian DOA vectors to determine whether an event class is active. In contrast to EINV2, the ACCDOA representation merges SED and DOA branches into a single branch, decreasing the model parameters and avoiding the necessity of balancing the loss measuring on the SED task and the DOA task.

The ACCDOA representation can not detect homogenous overlaps. Therefore, multi-ACCDOA which still contains a single branch and combines class-wise output format and track-wise output format, is proposed to overcome the problem [8]. While each track in the track-wise output format of EINV2 only detects one event class and a corresponding location, each track in the multi-ACCDOA predicts activities and corresponding locations of all target classes. Auxiliary duplicating permutation invariant training (ADPIT) is also proposed to train each track of the multi-ACCDOA with original targets and duplicated targets, enabling each track to regard the same target as the single one. The multi-ACCDOA representation is shown in Fig. 1. Its outputs are track-wise and class-wise Cartesian DOA vectors. Each vector length indicates the activity of the event. Besides the activity threshold, multi-ACCDOA employs angle thresholds to determine whether the predicted objects are the same or different.

3. META-SELD

3.1. The SELD model

Without loss of generality, in this study, we adopt a simple Convolutional Recurrent Neural Network (CRNN) as our network, which is similar to the baseline of Task 3 of DCASE 2022 Challenge [2] but with ACCDOA format. The network has three convolution blocks followed by a one-layer bidirectional gated recurrent unit (BiGRU). The network takes the concatenation of log-mel spectrograms and intensity vectors as input and predicts active sound events with corresponding Cartesian DOA vectors for each time step. The network architecture of CRNN is shown in Table 1.

Table 1: The network architecture of CRNN

Log-mel spectrogram & Intensity vectors
(Conv2d 3×3 @ 32, BatchNorm2d, ReLU) $\times$ 2, Avg Pooling 2×2
(Conv2d 3×3 @ 64, BatchNorm2d, ReLU) $\times$ 2, Avg Pooling 2×2
(Conv2d 3×3 @ 128, BatchNorm2d, ReLU) $\times$ 2, Avg Pooling 2×2
(Conv2d 3×3 @ 256, BatchNorm2d, ReLU) $\times$ 2, Avg Pooling 1×2
Global average pooling @ frequency
1-layer BiGRU of 128 hidden size, 256×39 linear layer, Tanh
Mean Square Error

3.2. Meta-SELD training

Given a model represented by a parameterized function f_Θ with parameters Θ, MAML [14] learns the initial parameters Θ_0 from general tasks $\mathcal{T}_i$ sampled from the training set $\mathcal{D}_{\texttt{train}}$ and is expected to perform well on unseen tasks from the test set $\mathcal{D}_{\texttt{test}}$ after a few iterations of parameters update with a small number of samples from the corresponding task. These initial parameters are very sensitive to being further optimized on a specific task. Each task $\mathcal{T}_i$ consists of a labeled support set $\mathcal{S}_i$ of K samples and a labeled query set $\mathcal{Q}_i$ of Q samples. A new task is expected to be quickly adapted with K samples, which is known as K-shot learning. The loss function of MAML is defined as

$$\mathcal{L} = \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_\Theta) \tag{1}$$

where $p(\mathcal{T})$, which is sampled from $\mathcal{D}_{\texttt{train}}$, is a distribution over tasks that we want our model to be able to adapt to. In contrast to supervised deep learning methods, the objective of which is to find optimal parameters to minimize the loss function across all training samples, MAML tries to find generalized initial parameters for different tasks. MAML will then update the initial parameters after several iterations of training on data of new tasks.

There are two groups of parameters in the MAML algorithm, meta-parameters and adapt-parameters. In the meta-training phase, MAML starts with randomly initialized meta-parameters Θ and then adapts to a new specific task $\mathcal{T}_i$ with several update iterations using $\mathcal{S}_i$. The meta-parameters Θ become adapt-parameters Θ'_i:

$$\Theta'_i = \Theta - \alpha \nabla_\Theta \mathcal{L}_{\mathcal{T}_i}(f_\Theta, \mathcal{S}_i) \tag{2}$$

where α is the adaptation learning rate for adapt-parameters updates. After updates across a batch of tasks, the meta-parameters are updated as:

$$\Theta = \Theta - \beta \nabla_\Theta \sum_{\mathcal{T}_i} \mathcal{L}_{\mathcal{T}_i}\left(f_{\Theta'_i}, \mathcal{Q}_i\right) \tag{3}$$

Algorithm 1 Meta-training of MAML for Meta-SELD

Require: Distribution over all rooms $p(\mathcal{T})$, adaptation step size α, meta step size β
1: randomly initialize meta-parameters Θ
2: **while** not done **do**
3: Sample a batch of rooms $\mathcal{T}_i \sim p(\mathcal{T})$
4: **for** each room $\mathcal{T}_i$ **do**
5: Sample disjoint examples $(\mathcal{S}_i, \mathcal{Q}_i)$ from $\mathcal{T}_i$
6: Let $\Theta_{i,0} \leftarrow \Theta$
7: **for** gradient descent step $j := 0$ **to** $N-1$ **do**
8: Perform gradient descent to update adapt-parameters: $\Theta_{i,j+1} \leftarrow \Theta_{i,j} - \alpha \nabla_{\Theta_i} \mathcal{L}_{\mathcal{T}_i}\left(\Theta_{i,j}, \mathcal{S}_i\right)$
9: **end for**
10: Compute $\mathcal{L}_{\mathcal{T}_i}(f_{\Theta_{i,N}}, \mathcal{Q}_i)$
11: **end for**
12: Perform gradient descent to update meta-parameters: $\Theta \leftarrow \Theta - \beta \nabla_{\Theta} \sum_{\mathcal{T}_i} \mathcal{L}_{\mathcal{T}_i}\left(f_{\Theta_{i,N}}, \mathcal{Q}_i\right)$
13: **end while**

where β is the meta step size. The loss $\mathcal{L}_{\mathcal{T}_i}$ is calculated by the parameterized function $f_{\Theta'_i}$ on the query set $\mathcal{Q}_i$. After updating Θ on the query set, Θ will be used as initial parameters for the following meta-training steps.

We aim to adapt to an unseen environment with K samples (K-shot). The objective of MAML is to find optimal initial parameters across several tasks, so we need to construct a set of tasks from the training set $\mathcal{D}_{\texttt{train}}$. $\mathcal{D}_{\texttt{train}}$ is split according to the different recording rooms. Audio clips recorded in different rooms belong to different tasks. We first sample a batch of tasks from all tasks and then sample $K+Q$ samples in each task, where K samples for a support set $\mathcal{S}_i$ and Q samples for a query set $\mathcal{Q}_i$. The overall training procedure of MAML is summarized in Algorithm 1. Step 8 in Algorithm 1 is an inner-loop update for adapt-parameters, while Step 12 is outer-loop updates for meta-parameters.

3.3. Meta-SELD test

In the meta-testing phase, a specific unseen task $\mathcal{T}_j^{\texttt{test}}$ created using $\mathcal{D}_{\texttt{test}}$ is used. $\mathcal{T}_j^{\texttt{test}}$ consists of a labeled support set $\mathcal{S}_j^{\texttt{test}}$ of K samples, and an unlabeled query set $\mathcal{Q}_j^{\texttt{test}}$ of Q samples. After training the model using well-trained parameter Θ from the meta-training phase as the initial parameters on $\mathcal{S}_j^{\texttt{test}}$, we get updated parameters ${\Theta_j}'$. We then use $f_{\Theta'_j}$ to evaluate on $\mathcal{Q}_j^{\texttt{test}}$.

The meta processes for testing and training are slightly different. Similar to the training, the test set $\mathcal{D}_{\texttt{test}}$ is split according to the recording room of each audio clip. For clips of each room, we also chose K samples for meta-test support set $\mathcal{S}_j^{\texttt{test}}$ and all remaining samples for meta-test query set $\mathcal{Q}_j^{\texttt{test}}$. After N iterations of parameters update on $\mathcal{S}_j^{\texttt{test}}$, the meta-parameters Θ are updated to $\Theta_{j,N}$. The final performance is evaluated on $\mathcal{Q}_j^{\texttt{test}}$ with $f_{\Theta_{j,N}}$.

4. EXPERIMENTS

4.1. Dataset

There are 16 different recording rooms in total in the development set of the STARSS23 dataset, including nine recording rooms in *dev-train-set* and seven recordings rooms in *dev-test-set*. The development set of STARSS23, which contains roughly 7.5 hours of recordings, has less data than the development set in DCASE 2021, which contains roughly 13 hours of synthetic recordings [17]. Considering the complexity of the real-scene environment, we use additional datasets to improve the performance. We generated simulated data using the generator code provided by DCASE[1]. We synthesize multi-channel spatial recordings by convolving monophonic sound event examples with multi-channel Spatial Room Impulse Responses (SRIRs). Samples of sound events are selected from AudioSet [18] and FSD50K [19], based on the affinity of the labels in those datasets to target classes in STARSS23. PANNs [20] are then employed to clean the selection of the clips. We use pre-trained PANNs to infer these clips and select high-quality clips based on output probability above 0.8. We extracted SRIRs from the TAU Spatial Room Impulse Response Database (TAU-SRIR DB)[2], which contains SRIRs captured in 9 rooms at Tampere University. It was used for official synthetic datasets in DCASE 2019-2021 [1, 17, 21].

The 2700 1-minute audio clips that we synthesized using the abovementioned SRIRs from 9 rooms are used for $\mathcal{D}_{\texttt{train}}$, and all of *dev-set* of STARSS23, recorded in 16 rooms, are used for $\mathcal{D}_{\texttt{test}}$.

4.2. Experimental setup

The sampling rate of the dataset is 24 kHz. We extracted 64-dimensional log mel spectrograms from four-channel first-order ambisonics (FOA) signals with a Hanning window of 1024 points, and a hop size of 320. Each audio clip is segmented to a fixed length of five seconds with no overlap for training and inference.

In the meta-training phase, the training set and test set are divided into 9 tasks and 16 tasks, respectively, corresponding to 9 rooms and 16 rooms. We first sample a batch of rooms randomly and then sample a batch of examples from each of the rooms. The batch of samples of each room constructs a task, and a part of the samples are support samples while the remaining samples are query samples. The batch size of rooms and samples is 4 and 64, respectively. A batch of samples contains 30 support samples and 34 query samples. In the meta-test phase, we sort the audio clips according to the filename, and select the first 30 samples of recordings of each room as samples from the support set $\mathcal{S}_j^{\texttt{test}}$. The remaining samples of each room are as samples from the test set $\mathcal{Q}_j^{\texttt{test}}$. The AdamW optimizer is used for updates of meta-parameters of MAML, while the SGD optimizer is used to update adapt-parameters. The meta step size β begins with 0.001 in the first 100 epochs out of 150 epochs in total and is then decreased by 10% every 20 epochs. The adaptation step size and the number of update iterations are always kept at 0.01 and 5, respectively.

To demonstrate the effectiveness of Meta-SELD, we compare Meta-SELD with the fine-tuning method from the pre-trained SELD model. Firstly, we train a SELD model with AdamW optimizer in $\mathcal{D}_{\texttt{train}}$ from scratch. The learning rate is 0.0003 for the first 70 epochs and then decreases to 0.00003 for the following 20 epochs. Secondly, we initialize the parameters from the previously trained SELD model and then use $\mathcal{S}_i^{\texttt{test}}$ and $\mathcal{Q}_i^{\texttt{test}}$ as the training set and the test set of the i-th room to fine-tune. Similar to the process of the adapt-parameters updates in MAML, the SGD optimizer with a step size of 0.01 and update iterations of 5 are used for fine-tuning.

A joint metric of localization and detection [22, 23] is used: location-dependent F-score ($\mathrm{F}_{\leq 20^\circ}$) and error rate ($\mathrm{ER}_{\leq 20^\circ}$), and class-dependent localization recall ($\mathrm{LR}_{\mathrm{CD}}$) and localization error

[1] https://github.com/danielkrause/DCASE2022-data-generator
[2] https://zenodo.org/record/6408611

Table 2: The performance of the Meta-SELD and fine-tuning methods from pre-trained SELD models. Both two methods are evaluated in $\mathcal{Q}_i^{\texttt{test}}$. Note that *overall* scores of the fine-tuning method and Meta-SELD compute the fast adaptation performance of each individual room and then micro-average.

Room	$\mathrm{ER}_{20^\circ} \downarrow$			$\mathrm{F}_{20^\circ} \uparrow$			$\mathrm{LE}_{\mathrm{CD}} \downarrow$			$\mathrm{LR}_{\mathrm{CD}} \uparrow$			$\mathcal{E}_{\mathtt{SELD}} \downarrow$		
	Pre-train	Fine-tune	Meta	Pre-train	Fine-tune	Meta	Pre-train	Fine-tune	Meta	Pre-train	Fine-tune	Meta	Pre-train	Fine-tune	Meta
fold3_room4	0.624	**0.574**	0.603	**44.5%**	40.4%	29.8%	17.8°	**17.6°**	21.5°	**64.6%**	61.2%	54.4%	**0.408**	0.414	0.470
fold3_room6	0.639	0.607	**0.594**	38.0%	**40.5%**	40.4%	18.0°	**17.2°**	17.4°	**65.3%**	63.8%	61.1%	0.427	**0.415**	0.419
fold3_room7	0.610	**0.606**	0.660	**31.1%**	30.7%	20.8%	23.6°	24.1°	**22.5°**	59.9%	**60.5%**	48.3%	0.458	**0.457**	0.523
fold3_room9	0.673	**0.601**	0.608	43.7%	46.6%	**47.5%**	19.1°	18.6°	**18.3°**	**78.7%**	78.2%	73.3%	0.389	**0.364**	0.375
fold3_room12	0.685	**0.659**	0.689	28.0%	29.8%	**33.0%**	26.8°	**26.1°**	33.3°	43.1%	43.6%	**46.3%**	0.531	**0.518**	0.520
fold3_room13	0.650	0.599	**0.594**	37.7%	**39.4%**	36.1%	17.5°	16.9°	**15.9°**	**50.9%**	48.8%	37.1%	0.465	**0.453**	0.488
fold3_room14	0.633	**0.582**	0.613	**40.2%**	37.4%	28.6%	**23.2°**	23.7°	24.8°	**55.3%**	54.0%	47.2%	0.452	**0.450**	0.498
fold3_room21	0.757	0.750	**0.735**	19.3%	**21.6%**	18.9%	20.5°	**18.9°**	20.6°	39.3%	31.4%	**43.8%**	0.571	0.581	**0.556**
fold3_room22	0.850	0.818	**0.800**	11.4%	12.8%	**16.7%**	31.6°	29.5°	**29.0°**	45.6%	43.8%	**48.8%**	0.614	0.604	**0.577**
fold4_room2	0.809	0.774	**0.753**	6.2%	8.2%	**15.4%**	47.8°	41.3°	**33.0°**	72.4%	72.4%	**75.7%**	0.572	0.550	**0.506**
fold4_room8	0.716	0.716	**0.702**	31.7%	**33.6%**	30.7%	22.5°	**21.0°**	23.2°	**54.0%**	49.4%	49.4%	**0.496**	0.501	0.507
fold4_room10	0.792	0.708	**0.651**	36.3%	**41.7%**	35.8%	23.8°	21.5°	**20.2°**	66.1%	72.0%	**78.2%**	0.475	0.423	**0.406**
fold4_room15	0.582	0.563	**0.539**	33.3%	33.5%	**43.4%**	16.5°	**15.5°**	19.3°	42.8%	42.6%	**59.0%**	0.478	0.472	**0.406**
fold4_room16	0.601	**0.584**	0.607	39.8%	**40.5%**	34.3%	21.7°	21.9°	**21.6°**	**55.1%**	54.9%	48.7%	0.443	**0.438**	0.474
fold4_room23	0.813	0.746	**0.676**	25.4%	26.5%	**31.8%**	26.2°	**24.9°**	25.8°	40.4%	43.6%	**47.3%**	0.575	0.546	**0.507**
fold4_room24	0.828	**0.779**	0.782	26.2%	25.7%	**30.8%**	**19.4°**	19.7°	24.4°	41.0%	**43.6%**	42.7%	0.566	0.549	**0.546**
Overall	0.707	0.677	**0.672**	23.0%	24.2%	**26.0%**	22.8°	22.3°	**21.9°**	39.5%	40.2%	**41.0%**	0.552	0.539	**0.531**

($\mathrm{LE}_{\mathrm{CD}}$). $\mathrm{F}_{\leq 20^\circ}$ and $\mathrm{ER}_{\leq 20^\circ}$ consider true positives predicted under a spatial threshold 20° from the ground truth. $\mathrm{LE}_{\mathrm{CD}}$ and $\mathrm{LR}_{\mathrm{CD}}$ are computed for localization predictions in the case that the types of sound events are predicted correctly. A macro-average of $\mathrm{F}_{\leq 20^\circ}$, $\mathrm{LR}_{\mathrm{CD}}$ and $\mathrm{LE}_{\mathrm{CD}}$ is used.

We use an aggregated SELD metric which was computed as

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

4.3. Experimental results

Table 2 shows the performance of the Meta-SELD method compared with the fine-tuning method from the pre-trained SELD models. The pre-trained SELD models are trained without using samples from $\mathcal{D}_{\texttt{test}}$.

According to the last row of Table 2, the *overall* score, which is a micro average across all rooms, shows that all of $\mathrm{ER}_{\leq 20^\circ}$, $\mathrm{F}_{\leq 20^\circ}$, $\mathrm{LE}_{\mathrm{CD}}$, and $\mathrm{LR}_{\mathrm{CD}}$ are improved using Meta-SELD compared with the fine-tuning method. We observe a drop in $\mathcal{E}_{\mathtt{SELD}}$ in fold3_room4 and fold4_room8 even though some new samples of unseen environments are used for training. This may be due to the fact that the new samples do not have valid information for training. We also observe the Meta-SELD method improves $\mathcal{E}_{\mathtt{SELD}}$ by a large margin in fold3_room22, fold4_room2, and fold4_room23 where the pre-trained model has poor performance across all rooms. Specifically, $\mathrm{ER}_{\leq 20^\circ}$, $\mathrm{F}_{\leq 20^\circ}$, and $\mathrm{LR}_{\mathrm{CD}}$ of fold3_room22 and fold4_room23 outperform other methods. Meta-SELD mainly improves the performance of SED in fold3_room22 and fold4_room23. All metrics of fold4_room2 are improved in Meta-SELD compared with the fine-tuning method, especially in DOA estimation. In fold4_room2, all of the pre-trained model, the fine-tuning method, and Meta-SELD achieve $\mathrm{LR}_{\mathrm{CD}}$ of over 70%, but $\mathrm{LE}_{\mathrm{CD}}$ of three methods is always high compared with $\mathrm{LE}_{\mathrm{CD}}$ of other rooms. Meta-SELD decreases 14.8° and 8.3° of $\mathrm{LE}_{\mathrm{CD}}$ compared with the pre-trained model and the fine-tuning method in fold4_room2, hence directly leading to the increase of $\mathrm{F}_{\leq 20^\circ}$ and the decrease of $\mathrm{ER}_{\leq 20^\circ}$. However, performance degradation happens in fold3_room4, fold3_room7, fold3_room14, and fold4_room16, where Meta-SELD has the worst metric scores. There is no significant change in $\mathrm{LE}_{\mathrm{CD}}$, and the decline in SED performance is the main factor. One of the possible reasons for this observation could be that there are some conflicts in optimizing Meta-SELD across a batch of rooms.

Experimental results demonstrate that Meta-SELD can find better initial parameters across a batch of tasks than the fine-tuning method, especially in rooms where the pre-trained model and the fine-tuning method perform worse. Meta-SELD reduces the risk of overfitting when using a small number of samples, which usually happens in the fine-tuning method.

5. CONCLUSION

In this paper, we presented Meta-SELD, which employed Model-Agnostic Meta-Learning (MAML) to the sound event localization and detection task to achieve fast adaptation to unseen environments. The method only utilizes a small number of samples and a few update iterations of training. We use the STARSS23 dataset and synthesized 2700 1-minute samples that are convolved using monophonic sound event clips with multi-channel spatial room impulse responses. The sound event clips are extracted from FSD50K and AudioSet and are further filtered by the PANNs model through a probability threshold. The SRIRs used are from TAU-SRIR DB. Our methods are trained on synthetic datasets and evaluated on all development sets of the STARSS23 dataset. Audio clips recorded from the same room or synthesized using SRIRs collected from the same room are regarded as the same task for MAML. The experimental results show that the Meta-SELD method improves $\mathcal{E}_{\mathtt{SELD}}$ significantly in those rooms where both the pre-trained model and the fine-tuning method perform unsatisfactorily. The overall score demonstrates that the Meta-SELD method outperforms the fine-tuning method on average.

6. ACKNOWLEDGEMENT

This work was supported in part by Grant "XJTLU RDF-22-01-084", UK Engineering and Physical Sciences Research Council (EPSRC) Grant EP/T019751/1 "AI for Sound (AI4S)".

7. REFERENCES

[1] S. Adavanne, A. Politis, and T. Virtanen, "A multi-room reverberant dataset for sound event localization and detection," in *Proc. DCASE 2019 Workshop*, 2019, pp. 10–14.

[2] 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. DCASE 2022 Workshop*, 2022, pp. 161–165.

[3] 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, pp. 34–48, 2018.

[4] Y. Cao, T. Iqbal, Q. Kong, Y. Zhong, W. Wang, and M. D. Plumbley, "Event-independent network for polyphonic sound event localization and detection," in *Proc. DCASE 2020 Workshop*, 2020, pp. 11–15.

[5] 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 2021*, 2021, pp. 885–889.

[6] J. Hu, Y. Cao, M. Wu, Q. Kong, F. Yang, M. D. Plumbley, and J. Yang, "A track-wise ensemble event independent network for polyphonic sound event localization and detection," in *Proc. IEEE ICASSP 2022*, 2022, pp. 9196–9200.

[7] 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. IEEE ICASSP 2021*. IEEE, 2021, pp. 915–919.

[8] 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 2022*, 2022, pp. 316–320.

[9] J. Hu, Y. Cao, M. Wu, Q. Kong, F. Yang, M. D. Plumbley, and J. Yang, "Sound event localization and detection for real spatial sound scenes: Event-independent network and data augmentation chains," in *Proc. DCASE 2022 Workshop*, 2022, pp. 46–50.

[10] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–34, 2020.

[11] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 5149–5169, 2021.

[12] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *Proc. ICLR 2019*, 2019.

[13] N. Gao, H. Ziesche, N. A. Vien, M. Volpp, and G. Neumann, "What matters for meta-learning vision regression tasks?" in *Proc. CVPR 2022*, 2022, pp. 14 776–14 786.

[14] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. ICML 2017*, 2017, pp. 1126–1135.

[15] S.-F. Huang, C.-J. Lin, D.-R. Liu, Y.-C. Chen, and H.-y. Lee, "Meta-TTS: Meta-learning for few-shot speaker adaptive text-to-speech," *IEEE/ACM Trans. on Audio, Speech, and Lang. Process.*, vol. 30, pp. 1558–1571, 2022.

[16] M. Barhoush, A. Hallawa, A. Peine, L. Martin, and A. Schmeink, "Localization-driven speech enhancement in noisy multi-speaker hospital environments using deep learning and meta learning," *IEEE/ACM Trans. on Audio, Speech, and Lang. Process.*, vol. 31, pp. 670–683, 2022.

[17] 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. DCASE 2021 Workshop*, 2021, pp. 125–129.

[18] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE ICASSP 2017*, 2017, pp. 776–780.

[19] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: An open dataset of human-labeled sound events," *IEEE/ACM Trans. on Audio, Speech, and Lang. Process.*, vol. 30, pp. 829–852, 2021.

[20] 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. on Audio, Speech, and Lang. Process.*, vol. 28, pp. 2880–2894, 2020.

[21] 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 2020 Workshop*, 2020, pp. 165–169.

[22] A. Mesaros, S. Adavanne, A. Politis, T. Heittola, and T. Virtanen, "Joint measurement of localization and detection of sound events," in *Proc. IEEE WASPAA 2019*, 2019, pp. 333–337.

[23] 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. on Audio, Speech, and Lang. Process.*, vol. 29, pp. 684–698, 2020.