AN IMPROVED EVENT-INDEPENDENT NETWORK FOR POLYPHONIC SOUND EVENT LOCALIZATION AND DETECTION

Yin Cao[1]*, Turab Iqbal*[1]*, Qiuqiang Kong*[2]*, Fengyan An*[3]*, Wenwu Wang*[1]*, Mark D. Plumbley*[1]

[1]Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, UK
{yin.cao, t.iqbal, w.wang, m.plumbley}@surrey.ac.uk
[2]ByteDance Shanghai, China, kongqiuqiang@bytedance.com
[3]Qingdao University of Technology, China, anfy@qut.edu.cn

ABSTRACT

Polyphonic sound event localization and detection (SELD), which jointly performs sound event detection (SED) and direction-of-arrival (DoA) estimation, has better real-world applicability than separate SED or DoA estimation. It detects the type and occurrence time of sound events as well as their corresponding DoA angles simultaneously. We study the SELD task from a multi-task learning perspective. Two open problems are addressed in the paper. Firstly, to detect overlapping sound events of the same type but with different DoAs, we propose to use a trackwise output format and solve the accompanying track permutation problem with permutation-invariant training. Multi-head self-attention is further used to separate tracks. Secondly, a previous finding is that, by using hard parameter-sharing, SELD suffers from a performance loss compared with learning the subtasks separately. This is solved by a soft parameter-sharing scheme. We term the proposed method as Event Independent Network V2 (EINV2), which is an improved version of our previously-proposed method and an end-to-end network for SELD. We show that our proposed EINV2 for joint SED and DoA estimation outperforms previous methods by a large margin. In addition, a single EINV2 model with a VGG-style architecture has comparable performance to state-of-the-art ensemble models. Source code is available.

Index Terms— Sound event localization and detection, direction of arrival, event-independent, permutation-invariant training, multi-task learning.

1. INTRODUCTION

Sound source localization is a challenging research topic [1], with applications in area such as moving robots, scene visualization systems, and smart homes [2]. Sound event localization and detection (SELD) estimates the locations of sound sources and detects the corresponding types and occurrence time of the sound events.

For sound event detection (SED), learning-based methods [3], which learn models on a dataset, have recently achieved state-of-the-art performance [4]. However, for DoA estimation, parametric methods, [5] that use traditional signal processing algorithms, and learning-based methods, seem to have different strengths. Parametric methods do not need a training dataset, but their generalization ability is poor and the number of sources needs to be known *a priori* [5]. Learning-based methods, on the other hand, need a labeled dataset, but have a better capacity for adapting different complex environments. In this paper, we focus on learning-based methods.

SELD was first introduced in Task 3 of the 2019 Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge, which used the TAU Spatial Sound Events 2019 dataset [6,7]. A more challenging dataset with moving sources was released in Task 3 of the 2020 DCASE Challenge [8]. In the datasets for both challenges, there can be up to two overlapping events. For SELD, we previously introduced a two-stage method which detects the sound event first and then transfers the learned representations to learn direction-of-arrival (DoA) features [9]. While it achieved a good ranking in DCASE 2019, the method intrinsically treats SELD as two separate tasks, without continuously utilizing the essential interactions between SED and DoA estimation. In addition, due to its output format, it is unable to detect different sound events of the same type but with different DoAs. We call this situation as *homogeneous overlap*.

In this paper, we focus on two open issues: the output format and joint SELD learning. We show that, given an effective joint SELD learning scheme, SED and DoA estimation can be trained together with mutual benefits. Source code is released[1].

Firstly, to detect the homogeneous overlap, the trackwise output format is investigated thoroughly. We proposed our Event-Independent Network using the trackwise output format to detect the homogeneous overlap [10]. The trackwise output format assumes that the output of the network has several tracks, each with at most one predicted event with a corresponding DoA. Different tracks can detect events of the same type with different DoAs, which means trackwise output format can detect the homogeneous overlap [10]. However, the trackwise output format introduces a *track permutation* problem, which is similar to the talker permutation problem in speaker separation [11, 12]. To address this, we investigate to adopt frame-level and chunk-level permutation-invariant training (PIT) to dynamically assign labels to the correct tracks during training. We also propose to use multi-head self-attention (MHSA) from transformers [13, 14] to separate the latent representations.

Secondly, we propose an improved joint-learning method with soft interactions between SED and DoA estimation for the trackwise output format. From a learning perspective, SELD can be considered as a Multi-Task Learning (MTL) problem [15]. MTL is typically done with either hard parameter-sharing (PS) or soft PS. Hard PS means that subtasks use the same high-level feature layers, while soft PS means that different subtasks use feature layers of their own, connections between those different feature layers exist. From previous research [7,9], it is found that hard PS used in SELDnet or the domain adaptation, which inductively transfers the learned representations from task to task, as used in our previous two-stage method, does not learn an optimal model. Instead, we propose to adopt a soft PS strategy between SED and DoA estimation to solve problems. In

[1]https://github.com/yinkalario/EIN-SELD

particular, by using concepts inspired by the cross-stitch module [16], we incorporate soft PS for both high-level feature layers and the MHSA that is used for track separation.

We propose a method that combines the trackwise output format and the soft PS scheme. We call this method the Event-Independent Network V2 (EINV2). To the best of our knowledge, it is the first time that SELD has been discussed from an effective MTL perspective. We will show that the proposed method outperforms previous methods by a large margin, and a single EINV2 model using a simple VGG-style architecture gives comparable performance to the state-of-the-art ensemble models on Task 3 of the 2020 DCASE Challenge.

The rest of the paper is arranged as follows. Section 2 reviews related works. Section 3 introduces the proposed method. Section 4 shows experimental results. Section 5 summarizes conclusions.

2. RELATED WORKS

2.1. Sound Event Localization and Detection

SELD has received wide attention since Task 3 of 2019 DCASE Challenge [6, 7, 17, 18]. A new TAU-NIGENs Spatial Sound Events dataset with moving sound sources that promotes research in this area was recently released [8, 19]. Adavanne et al. proposed SELDnet, where SED and DoA estimation shares high-level feature layers [7]. We proposed a two-stage method by means of domain adaptation [9]. Grondin et al. used a CRNN on pairs of microphones to perform SELD [20]. Nguyen et al. proposed to use a sequence matching network to align SED and DoA predictions [21]. Mazzon et al. proposed a spatial-augmentation method by rotating channels [22]. Shimada et al. proposed an ACCDOA method to train only on location information but using modules of ACCDOA vectors as SED activations [23]. In this paper, an improved Event-Independent Network V2 using trackwise output format and a soft PS scheme is proposed.

2.2. Multi-Task Learning

MTL has been successfully applied to almost all areas of machine learning [15, 24, 25], such as natural language processing [26] and computer vision [27]. MTL is inherently a multi-objective problem with conflicts existing among tasks [28]. A weighted linear combination of per-task losses is a common compromised solution. However, this simple solution may be invalid and make MTL detrimental when tasks heavily compete. Therefore, how features are shared among tasks is an essential problem. Some methods are investigated, e.g. fully-adaptive feature sharing [29], joint many-task model [30], Panoptic Feature Pyramid Networks [31], etc. Previous experimental results show that SED and DoA share some common features but also compete, which makes the hard PS used by SELDnet [7] a suboptimal solution. In this paper, a soft PS method is proposed for EINV2.

3. THE PROPOSED METHOD

This section introduces the proposed EINV2. It includes the trackwise output format, PIT for solving the track permutation problem, multi-head self-attention for separating tracks, and soft parameter-sharing.

3.1. SELDnet Output Format

We first review the SELDnet output format [7]. An illustration of the output format is given in Fig. 1. Mathematically the SELDnet output format is defined as

$$\boldsymbol{Y}_{\text{SELDnet}} = \{(y_{\text{SED}}, y_{\text{DoA}}) \,|y_{\text{SED}} \in \mathbb{1}_{\mathbf{S}}^{K}, y_{\text{DoA}} \in \mathbb{R}^{K\times 3}\}, \quad (1)$$

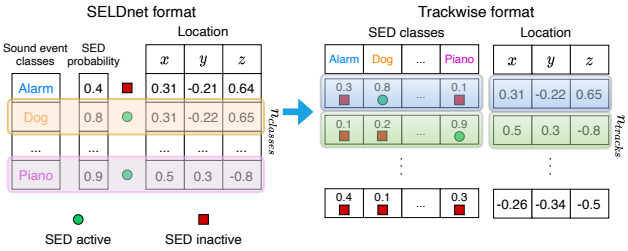


Fig. 1: Illustration of output formats, from SELDnet to Trackwise.

where y_{SED} and y_{DoA} are predictions for SED and DoA, respectively, $\mathbb{1}_{\mathbf{S}}^{K}$ is the one hot encoding for K classes, $\mathbf{S}$ is the set of sound event classes, and the number of dimensions of Cartesian coordinates is 3.

SELDnet predicts probabilities of all sound events and corresponding locations. A threshold is then used to binarize the probabilities of events. Binarized probabilities are used to activate corresponding sound events and locations. The number of predicted event types and locations are the same (their dimensions are both K), which means that there is only one location per predicted event.

The SELDnet output format cannot detect the same event with multiple locations, hence it is invalid when there is a homogeneous overlap. In addition, we found that when only a few types of sound events are active, locations of other inactive events tend to be very similar to locations of activated events. For instance, in Fig. 1, the SELDnet output format shows that the event "Alarm" is not active, but its location is very similar to the location of "Dog". Considering the fact that there is only a limited number of active sources at frame t, predicting locations for all of the event types, no matter if they are active or not, is unnecessary. Therefore, dimensions of location are redundant. Regression naturally fits when estimating continuous locations. Redundant dimensions of regression may increase the demand for the training-data size and the model capacity.

3.2. Trackwise Output Format

The trackwise output format was first proposed in our previous work [10] and is also shown in Fig. 1. It can be defined as

$$\boldsymbol{Y}_{\text{Trackwise}} = \{(y_{\text{SED}}, y_{\text{DoA}}) \,|y_{\text{SED}} \in \mathbb{1}_{\mathbf{S}}^{M\times K}, y_{\text{DoA}} \in \mathbb{R}^{M\times 3}\}, \quad (2)$$

where M is the number of tracks. Since K is the total sound events categories, $M \ll K$ in general.

Each track only detects one event and a corresponding location. The number of tracks M is the desired number of DoAs to detect. Hence, instead of estimating locations for all of the K events regardless of whether they are active or not, the trackwise output format only estimates M locations. This greatly reduces the demand for the model capacity and the required data size. In addition, the output format can now detect the homogeneous overlap.

The trackwise output format introduces a *track permutation* problem [10]. Since $M \ll K$, events are not always predicted in fixed tracks. At frame t, "Dog" can be predicted in track 1, but at next frame $t+1$, "Dog" can be predicted in track 2. The prediction of "Dog" is not fixed on track 1 or 2. This would result in a consequence that tracks do not "know" the correct ground truth during training. Permutation-invariant training is used to solve this problem.

3.3. Permutation-Invariant Training

Permutation-invariant training (PIT) was first proposed for speaker separation [11]. Our previous work discussed the benefit of using PIT for the trackwise output format [10]. Frame-level and chunk-level

PIT are both used in this paper. Here, chunk is a whole segment spanning from the start to the end of an event. Frame-level PIT assumes labels among frames are independently assigned, chunk-level PIT assumes labels that are within the same chunk of audio event are assigned to the same track. Let o denote the frame index t or the chunk index c. Given a permutation set $\mathbf{P}(o)$ consisting of all possible prediction-label pairs at index o, ground truth labels are assigned using all possible combinations in $\mathbf{P}(o)$. The lowest loss for each frame (tPIT) or for each chunk (cPIT) will be chosen to perform the back-propagation. The PIT loss can be defined as

$$\mathcal{L}^{PIT}(o) = \min_{\alpha \in \mathbf{P}(o)} \sum_{M} \{\ell_{\alpha}^{\text{SED}}(o) + \ell_{\alpha}^{\text{DoA}}(o)\}, \tag{3}$$

where $\alpha \in \mathbf{P}(o)$ is one of the possible permutation pairs.

3.4. Multi-Head Self-Attention

The MHSA from transformers [13] is used to separate tracks. A fixed absolute positional encoding is used before MHSA as:

$$\boldsymbol{P}_{(t,2i)} = 0.1 \sin\left(t/10^{8i/D_c}\right), \boldsymbol{P}_{(t,2i+1)} = 0.1 \cos\left(t/10^{8i/D_c}\right), \tag{4}$$

where t denotes the index of the time dimension, i denotes the index of the feature maps. Given an input $\boldsymbol{X} \in \mathbb{R}^{D_t \times D_{\text{in}}}$ with D_{in} denoting the input dimension, the Self-Attention (SA) can be written as:

$$\text{SA}\left(\boldsymbol{X}\right) := \text{softmax}\left((\boldsymbol{X}+\boldsymbol{P})\boldsymbol{W}_{\text{qry}}\boldsymbol{W}_{\text{key}}^{\top}(\boldsymbol{X}+\boldsymbol{P})^{\top}\right)\boldsymbol{X}\boldsymbol{W}_{\text{val}}, \tag{5}$$

where $\boldsymbol{W}_{\text{qry}}, \boldsymbol{W}_{\text{key}} \in \mathbb{R}^{D_{\text{in}} \times D_k}$ are learnable query and key matrices, respectively. D_k is the dimension of keys. $\boldsymbol{W}_{\text{val}} \in \mathbb{R}^{D_{\text{in}} \times D_{\text{out}}}$ is a learnable value matrix. In MHSA, it evenly splits D_{out} to N_h head, with each head has a dimension of D_h. MHSA can be expressed as:

$$\text{MHSA}(\boldsymbol{X}) := \underset{h \in [N_h]}{\text{concat}}\left[\text{ SA }_h(\boldsymbol{X})\right]\boldsymbol{W}_{\text{out}} + \boldsymbol{b}_{\text{out}} \tag{6}$$

where $\boldsymbol{W}_{\text{out}} \in \mathbb{R}^{N_h \cdot D_h \times D_{\text{out}}}$ and $\boldsymbol{b}_{\text{out}} \in \mathbb{R}^{D_{\text{out}}}$ are the projection matrix and the corresponding bias, respectively.

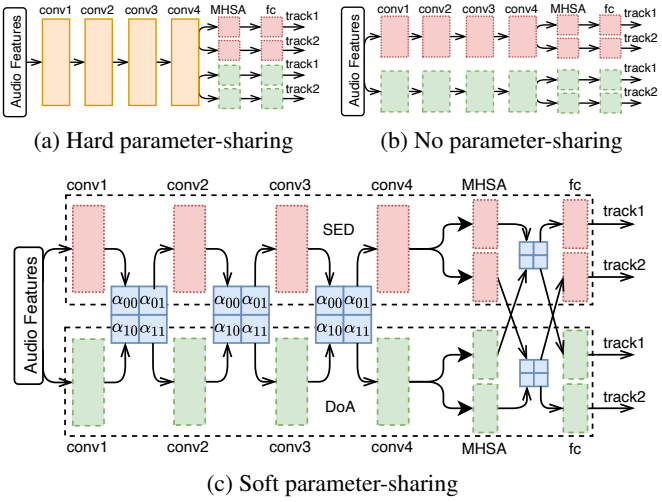
3.5. Parameter-Sharing Strategies

From an MTL perspective, joint SELD learning can be mutually beneficial [15, 24]. The explanation is three-fold: SED and DoA estimation have different noise patterns. When training a model on both SED and DoA estimation, the idea is to learn a good representation F that generalizes well through averaging both data-dependent or label-dependent noise patterns; some features R may be easy to learn for SED, while being hard to learn for DoA estimation. Using MTL, the model can eavesdrop to learn features R through SED; MTL uses several loss terms, which may act the same as loss regularizers. Hence, MTL can also reduce the model to overfit to SED or DoA estimation. Hard and soft PS are two typical methods to implement MTL. Hard PS means subtasks use the same feature layers, whereas soft PS means subtasks use their own feature layers with connections existing among those feature layers.

As shown in Fig. 2, three PS strategies are given for the trackwise output format. Fig. 2(a) shows hard PS for the trackwise output format. High-level representations are fully shared between two tasks. However, it is inevitable that as two different tasks, SED and DoA estimation conflict. Hard PS forces them to use the same features, which leads to a performance loss compared with learning the subtasks separately. Fig. 2(b) shows no PS with the trackwise output format. This method treats two tasks as separate ones. Since SED and DoA labels are connected, there are still weak connections between the two tasks when they are trained together. Fig. 2(c) shows soft PS with the trackwise output format. It uses soft PS between feature layers and MHSA layers in both SED and DoA estimation. Let D_c, D_t, D_f denote the dimensions of feature maps, time, and frequency, respectively. $\alpha_{ij} \in \mathbb{R}^{D_c}$ denotes learnable parameters. The new feature maps $(\hat{\boldsymbol{x}}^{\text{SED}}, \hat{\boldsymbol{x}}^{\text{DoA}}) \in \mathbb{R}^{D_c \times D_t \times D_f}$ can be calculated from the original feature maps $(\boldsymbol{x}^{\text{SED}}, \boldsymbol{x}^{\text{DoA}})$ as:

$$\left[\hat{\boldsymbol{x}}^{\text{SED}}, \hat{\boldsymbol{x}}^{\text{DoA}}\right]^{\top} = \boldsymbol{\alpha}\left[\boldsymbol{x}^{\text{SED}}, \boldsymbol{x}^{\text{DoA}}\right]^{\top}, \tag{7}$$

where $\boldsymbol{\alpha}$ is a 2×2 matrix that consists of elements α_{ij}, and $[\cdot]^{\top}$ denotes the transpose operation.



(a) Hard parameter-sharing (b) No parameter-sharing

(c) Soft parameter-sharing

Fig. 2: Different strategies of PS. Dotted-red is the SED task. Dashed-green is the DoA estimation task. Rectangular-blue boxes indicate soft connections between SED and DoA estimation.

Table 1: Event Independent Network V2

SED task, log-mel spectrogram		DoA estimation task, log-mel + intensity vector	
$(3 \times 3$ @ 64, BN, ReLU$) \times 2$, Pooling 2×2		$(3 \times 3$ @ 64, BN, ReLU$) \times 2$, Pooling 2×2	
Soft Parameter-Sharing, $\boldsymbol{\alpha}_{2\times2}$ @ 64			
$(3 \times 3$ @ 128, BN, ReLU$) \times 2$, Pooling 2×2		$(3 \times 3$ @ 128, BN, ReLU$) \times 2$, Pooling 2×2	
Soft Parameter-Sharing, $\boldsymbol{\alpha}_{2\times2}$ @ 128			
$(3 \times 3$ @ 256, BN, ReLU$) \times 2$, Pooling 1×2		$(3 \times 3$ @ 256, BN, ReLU$) \times 2$, Pooling 1×2	
Soft Parameter-Sharing, $\boldsymbol{\alpha}_{2\times2}$ @ 256			
$(3 \times 3$ @ 512, BN, ReLU$) \times 2$, Pooling 1×2		$(3 \times 3$ @ 512, BN, ReLU$) \times 2$, Pooling 1×2	
Global average pooling @ frequency		Global average pooling @ frequency	
Track 1, SED	Track 1, DoA	Track 2, SED	Track 2, DoA
(MHSA @ 512, 8 h) $\times$ 2	(MHSA @ 512, 8 h) $\times$ 2	(MHSA @ 512, 8 h) $\times$ 2	(MHSA @ 512, 8 h) $\times$ 2
Soft Parameter-Sharing, $\boldsymbol{\alpha}_{2\times2}$ @ 512		Soft Parameter-Sharing, $\boldsymbol{\alpha}_{2\times2}$ @ 512	
FC, 512 $\times$ 14, Sigmoid	FC, 512 $\times$ 3, Tanh	FC, 512 $\times$ 14, Sigmoid	FC, 512 $\times$ 3, Tanh
Binary Cross-Entropy	Mean Square Error	Binary Cross-Entropy	Mean Square Error
Frame-Level or Chunk-Level permutation-invariant Training			

3.6. Event-Independent Network V2

The proposed EINV2 combines the trackwise output format, PIT, MHSA, and soft PS. Table 1 shows the architecture of EINV2. The audio features used are log-mel spectrograms and intensity vectors in mel-space [10]. The trackwise output format can detect DoAs using only necessary dimensions. It can also detect the homogeneous overlap. PIT is used to solve the track permutation problem. MHSA is used to separate tracks. The soft PS is created between latent representations of SED and DoA estimation subtasks, the network can decide what useful information to exchange and what not to. This avoids a performance loss for both subtasks.

4. EXPERIMENTS

The dataset used is TAU-NIGENS Spatial Sound Events 2020 [8]. It consists of 14 types of sound events with continuous DoA angles

Table 2: Format comparison.

Method	SED_{only} F	DoA_{only} LE
SELDnet	**0.839**	17.21
$\text{Track}_{\text{noPIT}}$	0.795	37.57
$\text{Track}_{\text{cPIT}}$	0.832	12.55
$\text{Track}_{\text{tPIT}}$	**0.838**	**12.39**

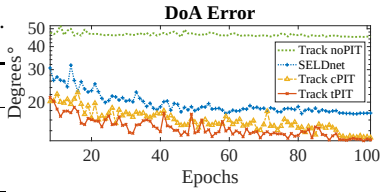


Fig. 3: Format performance.

Table 3: Performance on combined SELD task.

Methods	Output Format	$ER_{\leq 20^\circ}$	$F_{\leq 20^\circ}$	LE_{CD}	LR_{CD}
Baseline	SELDnet	0.720	37.4%	22.8°	60.7%
No-PS	SELDnet	0.399	67.5%	14.8°	83.8%
	Trackwise	0.340	73.7%	11.9°	83.8%
Hard-PS	SELDnet	0.414	65.9%	15.5°	82.4%
	Trackwise	0.339	73.9%	10.4°	81.3%
ACCDOA [23]	SELDnet	0.333	75.9%	10.3°	82.1%
Soft-PS	SELDnet	0.350	72.1%	13.1°	83.8%
EINV2	Trackwise	**0.299 ± 0.03**	**77.0 ± 0.3%**	**8.9 ± 0.2°**	**83.8 ± 0.2%**
No-PS-DA	Trackwise	0.323	75.8%	10.2°	83.1%
Hard-PS-DA	Trackwise	0.264	80.5%	7.9°	84.3%
ACCDOA-DA	SELDnet	0.293	80.0%	8.7°	84.4%
EINV2-DA	Trackwise	**0.233 ± 0.03**	**83.2 ± 0.2%**	**6.8 ± 0.2°**	**86.1 ± 0.3%**
ACCDOA Ensemble Model[2]	*SELDnet*	*0.25**	*83.2%**	*7.0°**	*86.2%**
Best Ensemble Model[2]	–	*0.20**	*84.9%**	*6.0°**	*88.5%**

spanning from $\phi \in [-180, 180)$ in azimuth and $\theta \in [-45, 45]$ in elevation. There are up to two overlapping events, hence M is set to 2. Four evaluation metrics are used [19], which are F-score $F_{\leq T^\circ}$, Error Rate $ER_{\leq T^\circ}$, localization error LE_{CD} and localization recall LR_{CD}. $F_{\leq T^\circ}$ and $ER_{\leq T^\circ}$ consider true positives predicted under a distance threshold $T = 20^\circ$ from the ground truth.

4.1. Hyper-Parameters

A 1024-point Hanning window with a hop size of 600 points is used for FFT. The number of mel bands is 256. Audio clips are segmented to have a fixed length of 4 seconds without overlapping for both training and test sets. The AdamW optimizer is used. The learning rate is set to 0.0005 for the first 90 epochs and is adjusted to 0.00005 for next 10 epochs that follows. The threshold for SED is 0.5 to binarize predictions. All experimental scores are trained on the development set and tested on the evaluation set. Final scores are averaged on five different trials.

4.2. Comparison of the trackwise and the SELDnet formats

The trackwise and the SELDnet formats are discussed first. To accurately compare the two formats, the single SED and DoA estimation tasks are tested individually. When testing SED alone, DoA predictions are set to be ground truth. A similar setup applies for DoA estimation. Both tasks use a single branch of EINV2 without soft PS shown in Table 1. Comparison results and the convergence behaviour of location error are shown in Table 2 and Fig. 3, respectively. $\text{Track}_{\text{noPIT}}$, $\text{Track}_{\text{cPIT}}$ and $\text{Track}_{\text{tPIT}}$ represents the trackwise output format without PIT, with chunk-level PIT, and with frame-level PIT, respectively. It can be seen that $\text{Track}_{\text{tPIT}}$ achieves the lowest location error with $\text{Track}_{\text{cPIT}}$ falling behind by 0.16°. SELDnet gets approximately 5° higher location error. $\text{Track}_{\text{noPIT}}$ does not converge. It is within expectations that the SELDnet output format has a higher location error than $\text{Track}_{\text{tPIT}}$, which reduces redundant dimensions for regression as discussed in section 3.2. When the trackwise output format is used, the track permutation problem prevents the correct labels from being assigned to the corresponding tracks. This can be elegantly solved using PIT. cPIT is more effective at tracking chunk-level events. However, the metrics used in this paper are frame-level, which may be the reason why $\text{Track}_{\text{tPIT}}$ is slightly better than $\text{Track}_{\text{cPIT}}$.

[2] https://bit.ly/31edoqC

4.3. Joint SELD Task

Several methods for the SELD task are compared. Results are shown in Table 3. No-PS, Hard-PS, and Soft-PS are the methods shown in Fig. 2. Among these methods, ACCDOA is the exception which only trains on the DoA task [23]. The proposed EINV2 method is a combination of Soft-PS and the trackwise output format. The second part with *DA* in the table shows the performance with rotation [22] and SpecAugment [32] data-augmentation methods.

It can be seen in the upper part of Table 3 that the trackwise output format achieves consistently lower LE_{CD} and better overall performance than the SELDnet output format for every method. When using the SELDnet output format, it shows No-PS outperforms Hard-PS, which indicates that hard parameter-sharing may be even worse than learning two separate tasks. Therefore, SED and DoA estimation can be detrimental to each other. It is a bit surprising that ACCDOA using the SELDnet output format outperforms No-PS and Hard-PS using the trackwise output format. This may be because ACCDOA turns the joint task into a single task, which eliminates the detriment caused by joint learning. However, ACCDOA still suffers from a performance loss in SED. It may be because ACCDOA uses mean-square-error instead of binary cross entropy as the loss to train SED. It is also challenging for ACCDOA to adapt to cases with different location radiuses. Among all compared methods, the proposed EINV2 shows the best performance without a performance compromise compared with learning the subtasks separately. All of its scores are better or equal to separate tasks. This indicates that by using an effective joint learning scheme, SELD can be trained together with mutual benefits.

It can be seen in the lower part of Table 3 that, when two data augmentation methods are applied, EINV2 outperforms other methods and is even comparable with the best ensemble models, which use complex models and more data augmentation methods. Note that EINV2 is a single model with basic VGG-style modules. It can be easily extended to some other networks, such as ResNet or DenseNet.

5. CONCLUSION

We have presented an improved Event Independent Network V2 for sound event localization and detection. It addresses two problems. First, a trackwise output format to detect sound events of the same type but with different DoAs. Both frame-level and chunk-level permutation-invariant training are used to solve the track permutation problem. Multi-head self-attention is used to separate predictions in different tracks. Second, a soft parameter-sharing scheme is adopted for joint SELD without a performance compromise compared with learning the subtasks separately. Experimental results show that the proposed EINV2 outperforms previous methods by a large margin. With a single VGG-style model used, EINV2 is comparable with the best ensemble models that use more data augmentation methods.

6. ACKNOWLEDGEMENT

This work was supported in part by EPSRC Grants EP/P022529/1, EP/N014111/1 “Making Sense of Sounds”, EP/T019751/1 “AI for Sound”, National Natural Science Foundation of China (Grant No. 11804365), and EPSRC grant EP/N509772/1, “DTP 2016-2017 University of Surrey”.

7. REFERENCES

[1] M. Brandstein and D. Ward, *Microphone Arrays: Signal Processing Techniques and Applications*, pp. 157–178, Springer, 2013.

[2] T. Virtanen, M. D. Plumbley, and D. Ellis, *Computational Analysis of Sound Scenes and Events*, pp. 3–12, Springer, 2018.

[3] Christopher M Bishop, *Pattern recognition and machine learning*, pp. 2–10, springer, 2006.

[4] L. Lin, X. Wang, H. Liu, and Y. Qian, "Guided Learning Convolution System for DCASE 2019 Task 4," in *DCASE Workshop 2019*, 2019, pp. 134–138.

[5] D. Pavlidi, A. Griffin, M. Puigt, and A. Mouchtaris, "Real-time Multiple Sound Source Localization and Counting Using a Circular Microphone Array," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2193–2206, 2013.

[6] S. Adavanne, A. Politis, and T. Virtanen, "A Multi-room Reverberant Dataset for Sound Event Localization and Detection," in *DCASE Workshop 2019*, 2019, pp. 10–14.

[7] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound Event Localization and Detection of Overlapping Sources Using Convolutional Recurrent Neural Networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 34–48, 2018.

[8] A. Politis, S. Adavanne, and T. Virtanen, "A Dataset of Reverberant Spatial Sound Scenes with Moving Sources for Sound Event Localization and Detection," in *DCASE Workshop 2020*, 2020.

[9] Y. Cao, Q. Kong, T. Iqbal, F. An, W. Wang, and M. Plumbley, "Polyphonic Sound Event Detection and Localization using a Two-Stage Strategy," in *DCASE Workshop 2019*, 2019, pp. 30–34.

[10] 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 *DCASE Workshop 2020*, 2020.

[11] D. Yu, M. Kolbæk, Z. Tan, and J. Jensen, "Permutation Invariant Training of Deep Models for Speaker-Independent Multi-Talker Speech Separation," in *Proc. on ICASSP*, 2017, pp. 241–245.

[12] M. Kolbæk, D. Yu, Z. Tan, and J. Jensen, "Multitalker Speech Separation with Utterance-level Permutation Invariant Training of Deep Recurrent Neural Networks," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1901–1913, 2017.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All you Need," in *Advances In Neural Information Processing Systems*, 2017, pp. 5998–6008.

[14] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient Transformers: A Survey," *arXiv preprint arXiv:2009.06732*, 2020.

[15] S. Ruder, "An Overview of Multi-Task Learning In Deep Neural Networks," *arXiv preprint arXiv:1706.05098*, 2017.

[16] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *Proc. on CVPR*, 2016, pp. 3994–4003.

[17] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for Polyphonic Sound Event Detection," *Applied Sciences*, vol. 6, no. 6, pp. 162, 2016.

[18] A. Politis, A. Mesaros, S. Adavanne, T. Heittola, and T. Virtanen, "Overview and Evaluation of Sound Event Localization and Detection in DCASE 2019," *arXiv preprint arXiv:2009.02792*, 2020.

[19] A. Mesaros, S. Adavanne, A. Politis, T. Heittola, and T. Virtanen, "Joint Measurement of Localization and Detection of Sound Events," in *Proc. on WASPAA*, 2019, pp. 333–337.

[20] F. Grondin, I. Sobieraj, M. D. Plumbley, and J. Glass, "Sound Event Localization and Detection Using CRNN on Pairs of Microphones," in *DCASE Workshop 2019)*, 2019, pp. 84–88.

[21] T. Nguyen, D. L. Jones, and W. S. Gan, "A Sequence Matching Network for Polyphonic Sound Event Localization and Detection," in *Proc. on ICASSP*. IEEE, 2020, pp. 71–75.

[22] L. Mazzon, Y. Koizumi, M. Yasuda, and N. Harada, "First Order Ambisonics Domain Spatial Augmentation for DNN-based Direction of Arrival Estimation," in *DCASE Workshop 2019*, 2019, pp. 154–158.

[23] K. Shimada, N. Takahashi, S. Takahashi, and Y. Mitsufuji, "Sound Event Localization and Detection Using Activity-Coupled Cartesian DOA Vector and RD3net," *arXiv preprint arXiv:2006.12014*, 2020.

[24] Y. Zhang and Q. Yang, "A Survey on Multi-task Learning," *arXiv preprint arXiv:1707.08114*, 2017.

[25] Y. Zhang, Y. Wei, and Q. Yang, "Learning to Multitask," in *Advances in Neural Information Processing Systems*, 2018, pp. 5771–5782.

[26] S. Subramanian, A. Trischler, Y. Bengio, and C. J. Pal, "Learning General Purpose Distributed Sentence Representations via Large Scale Multi-task Learning," in *Proc. on ICLR*, 2018.

[27] A. R. Zamir, A. Sax, W. Shen, L. J Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *Proc. on CVPR*, 2018, pp. 3712–3722.

[28] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," in *Advances in Neural Information Processing Systems*, 2018, pp. 527–538.

[29] Y. Lu, A. Kumar, S. Zhai, Y. Cheng, T. Javidi, and R. Feris, "Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification," in *Proc. on CVPR*, 2017, pp. 5334–5343.

[30] K. Hashimoto, C. Xiong, Y. Tsuruoka, and R. Socher, "A Joint Many-Task Model: Growing a Neural Network for Multiple NLP Tasks," in *Proc. on Empirical Methods in Natural Language Processing*, 2017, pp. 1923–1933.

[31] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic Feature Pyramid Networks," in *Proc. on CVPR*, 2019, pp. 6399–6408.

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