A Adaptive Few-Shot Learning Algorithm for Rare Sound Event Detection

Leilai Li, Jianzong Wang, Xiaoyang Qu, Chendong Zhao, Jing Xiao*

Ping An Technology (Shenzhen) Co., Ltd.

{lileilai446,wangjianzong347,quxiaoyang343,
zhaochendong343,xiaojing661}@pingan.com.cn

Abstract

Sound event detection is to infer the event by understanding the surrounding environmental sounds. Few-shot learning methods promise generating a well-trained model which is easily generalized when facing a new limit-data sound detect task without many training steps. Recent approaches have achieved significant results in this field. However, these approaches treat each support example independently ignoring the information of other examples (events) from the whole task. Because of this, most of previous methods are constrained to generate a same feature embedding facing all test-time tasks. An ideal model would construct a feature embedding adapted to the input task. In this work, we propose a novel task-adaptive module which is easily ported to any metric-based few-shot learning frameworks. The module could identify the task-relevant feature dimension. Incorporating our module improves performance considerably on two datasets over baseline methods, especially for the transductive propagation network. Such as +6.8% for *5-way 1-shot* accuracy on ESC-50, and +5.9% on noiseESC-50. We also investigate our approach in the domain-mismatch setting and achieve better results than previous methods.

Index Terms: Sound event detection, Few-shot learning, Data augmentation, Deep learning

1. Introduction

Automatic environmental sound event detection has received increasing attention in recent years [1]. It deals with audios detecting and classifying, which leads to multi-form applications in industry. Environmental sound is naturally different from other audios. It doesn't exhibit any stationary temporal patterns like phoneme in speech or rhythm in music. In contrast, sound event contains very complex temporal structure [2, 3, 4] that may be continues (e.g. rains), abrupt (e.g. thunder storm) or periodic (e.g. clock tick). Moreover, speech and music usually distribute on a relatively fixed frequency bandwidth, but sound event spans a wide frequency range where different sound's frequency may distribute in various patterns [5, 6]. The information contained in temporal patterns and frequency bins of the sound event could be massive [7].

To date, many deep-learning (DL) methods greatly improved detection performance [8, 9, 10, 11, 12]. However, they typically require large amounts of labeled data, which limits the generalization ability to limited-data tasks due to the annotation cost. These motivate the study of **Few-shot learning**. Meantime, this method has also been introduced in [13, 14, 15, 16, 17] which concerning rare sound event detection and achieved promising results. Few-shot model promises alleviating the problem of data deficit and cold start. It usually follows the episodic training strategy [13, 18], which considers an N-way K-shot (e.g 5 way, 1 shot means there're 5 classes, each contains 1 support example) classification task in each episode.

* Corresponding author: Jianzong Wang, jzwang@188.com.

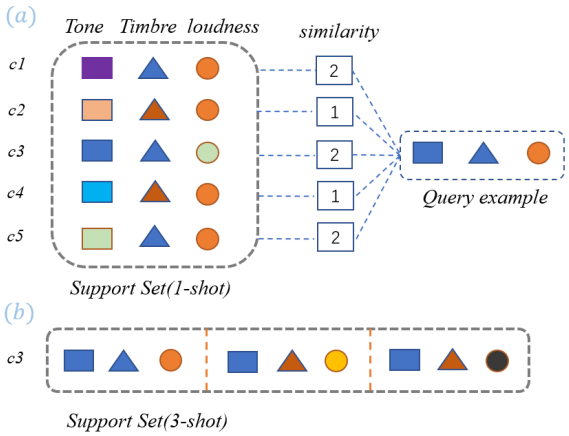

Figure 1: *An example illustrates the motivation. (a) defines a 5-way 1-shot task. There're three feature dimensions: Tone, Timbre and loudness. Different color means different value of the feature, same color adds the similarity score by 1; (b) In the k-shot (k=3) setting, all examples of class c3 share the same value of Tone even though their Timbre and loudness are different.*

All metric-based few-shot learning frameworks [16, 17, 19, 20, 21] compute similarity between each support (training) example and the query example independently, resulting in the correlation among support examples being missed. Figure 1 illustrates our motivation of the task-adaptive module. Each support example has different *Tone*, but may have same *Timbre* and *loudness* with others. During the similarity computation, the score between support example (c1, c3, c5) and the query example are all 2, making it hard to classify the query example. Moreover, in multi-shot setting like Figure 1 (b) shows, most of class c3 have the same *Tone* but various *Timbre* and *loudness*. Above all, the critical feature in this task should be *Tone*. Our task-adaptive module aims to integrate all support examples' information to value the commonality within per class and the uniqueness among all classes, thus to find the critical features.

Our contributions are as follows: (1) We introduce a feature encoder integrating attention mechanism to capture the temporal&channel context. In addition, a sound event-oriented data augmentaion strategy is introduced to alleviate the data-shortage problem. (2) We extend metric-based few-shot learning frameworks with a task-adaptive module to identify the uniqueness among classes and the commonality within per class of the whole support set. The task-adaptive module makes the feature-embedding process more effective. (3) We evaluated our model on two benchmarks. Results verify the superiority of our model over previous methods. Besides, our model also achieves better performance in the setting of domain mismatch.

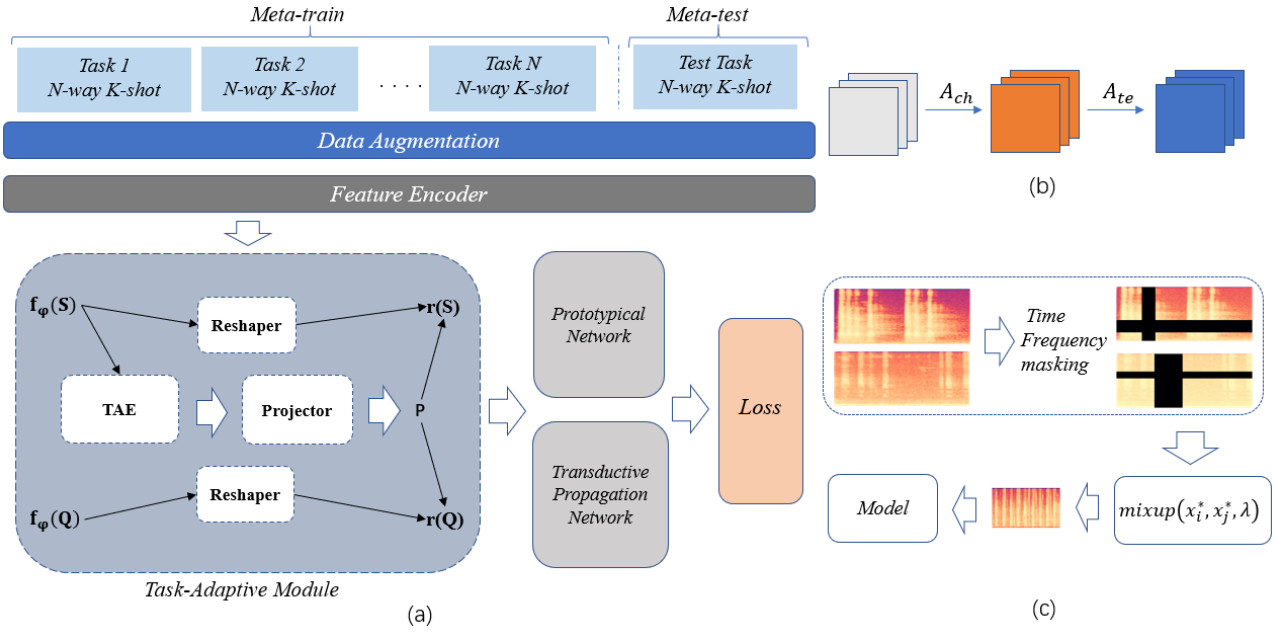

Figure 2: *(a). The overall framework of our model, it is composed of three parts: feature encoder, task-adaptive module and metric-based few-shot learning network. (b). The temporal&channel attention mechanism. A_{ch} is the channel attention, A_{te} is the temporal attention. (c). Data augmentation pipeline for the input log mel-spectrogram.*

2. Preliminary

2.1. Few-shot sound event detection

Recent approaches [9, 13, 15] integrate the prototypical networks [19] with graph neural networks [22] for few-shot sound event detection. Few-shot sound event detection aims to correctly classify unlabeled sounds with a few labeled examples.

Few-shot learning follows the episodic training paradigm that used in previous literature [19, 23]. Supposing there're two non-overlapping datasets of classes (events) $\boldsymbol{C}_{train}$ and $\boldsymbol{C}_{test}$, where $\boldsymbol{C}_{train} \cap \boldsymbol{C}_{test} = \emptyset$. There are also two procedures: meta-training and meta-testing. In one episode of training, we randomly sample N classes (a small subset) from $\boldsymbol{C}_{train}$ to construct support set $\mathcal{S}$ and query set $\mathcal{Q}$. A simple N-way K-shot task denotes as follow: $\mathcal{S}$ is denoted as $\mathcal{S} = \{x_1^1, ..., x_K^1, .., x_1^N, .., x_K^N\}$, where K is the number of samples in per class. The query set $\mathcal{Q} = \{x_1^*, ..., x_T^*\}$ contains various examples from the same N classes. Thus, there are NK samples in $\mathcal{S}$ and T samples in $\mathcal{Q}$. The support set and query set composed a multi-label classification task here. During the procedure of meta-testing, $\mathcal{S}$ and $\mathcal{Q}$ sampled from $\boldsymbol{C}_{test}$, and few-shot method is required to predict on query set (no label) given the support set (with label).

2.2. Metric-based learning methods

Few-shot learning methods can be divided into three branches: metric-based, optimization-based and data augmentation-based [24, 25, 26]. In this paper, we mainly focus on metric-based learning. The metric-based methods could be categorized into inductive and transductive, and two representative methods are prototypical network [19] and transductive propagation network (*TPN*) [17]. Given a N-way K-shot task $\mathcal{T} = (\mathcal{S}, \mathcal{Q})$, and the feature encoder f_φ decided by its internal parameter φ.

Prototypical network: this approach simply integrate NN-baseline into the end-to-end meta-learning framework. It takes the average of the learned representation of a few example for each class as class-wise representation, and then classifies an unlabeled instance by calculating the Euclidean distance between the input and the class-wise representations. $x_i^s \in \mathcal{S}$ and $x^q \in \mathcal{Q}$, and $\mathcal{M}$ is a pair-wise feature distance function. The distance measure is as follows:

$$f_{sim}(x^s, x^q; \mathcal{S}, \mathcal{Q}, \varphi) = \mathcal{M}(\frac{1}{K}\sum_{i=1}^{K} f_\varphi(x_i^s), f_\varphi(x^q)) \quad (1)$$

Transductive propagation network (TPN): this approach utilizes the entire test set for transductive inference, which is to consider the relationships among testset and thus predict them as a whole. Transductive inference has shown to outperform inductive methods [27, 28]. TPN propose to learn to propagate labels via episodic paradigm. During the propagation, a distance measurement and example-wise length scale parameter were adopted to obtain a proper neighborhood graph. After the graph construction, label propagation determines the labels of the query set. The distance measure is as follows:

$$f_{sim}(x_i, x_j; \mathcal{S}, \mathcal{Q}, \varphi, \phi) = exp(-\frac{1}{2}\mathcal{M}(\frac{f(x_i)}{\sigma_i}, \frac{f(x_j)}{\sigma_j})) \quad (2)$$

where ϕ is the parameters generating example-wise length-scale parameter (σ_i, σ_j). $x_i, x_j \in \mathcal{S} \cup \mathcal{Q}$.

3. Approach

3.1. Masked mixup

To avoid possible overfitting caused by limited trained data, we adopt time and frequency masking [29] and mix-up as the data augmentation strategy, which is simple but effective [30]. The strategy adopt multiple time and frequency masks on input spectrogram to generate multi-masked spectrograms and then randomly mixes two masked samples, increasing the diversity of samples by the way. As shown in Figure 2 (c), given a spectrogram x with T frames and F frequency bins. The first step is to use multiple time masking and frequency masking, generating M masked samples x^*. To be specific, $Mask$ assigns t consecutive time frames $[t_0 : t_0 + t)$ and f consecutive frequency bins $[f_0 : f_0 + f)$ value to 0. t is chosen from a uniform distribution from 0 to the time parameter τ, t_0 is chosen from $[0, T - t)$, f is chosen from a uniform distribution from 0 to the frequency

parameter υ, f_0 is chosen from $[0, F-f)$. Secondly, the mixup step conducts a convex combination of two randomly selected (x_i^*, y_i) and (x_j^*, y_j) from all masked samples:

$$x^* \leftarrow x \odot Mask \tag{3}$$

$$\tilde{x} = \lambda x_i^* + (1-\lambda) x_j^* \tag{4}$$

$$\tilde{y} = \lambda y_i + (1-\lambda) y_j \tag{5}$$

where y_i and y_j are one-hot encoded class labels, $(\tilde{x}, \tilde{y})$ being the new sample. $\lambda \in [0, 1]$ is acquired by sampling from a beta distribution $Beta(\alpha, \alpha)$ with α being a hyperparameter.

Feature encoder: After data augmentation, the samples first encoded by a ConvNet f_φ as same as [13], but the CNN layer replaced with a temporal&channel attention CNN layer [29] as shown in Figure 2 (b). The architecture of ConvNet contains five blocks and a fully-connected layer, and the first two block includes a 3×3 convolutional layer , a batch normalization layer, a ReLU activation layer and a 4×4 max-pooling layer, and the last block is same as the former except the last 1×1 max-pooling layer.

3.2. Task-adaptive module

This module contains three parts and leverages the $f_\varphi(.)$ as input, and outputs the task-adapted feature embedding, which will be passed to subsequent metric-based learning network.

3.2.1. Task-Adaptive-Extractor: commonality among class

TAE aims to find the commonality among all instances within a class. Denote the output shape from feature encoder f_φ as $(N \times K, m_1, w_1, h_1)$, where m_1, w_1, h_1 indicate the number of channel and spatial size respectively. *TAE* is defined as follows:

$$f_\varphi(\mathcal{S}) : (N \times K, m_1, w_1, h_1) \xrightarrow{TAE} o : (N, m_2, w_2, h_2) \tag{6}$$

where m_2, w_2, h_2 denote the output number of channel and spatial size. In this part, we first utilize a simple CNN layer to perform the dimension reduction. Then the samples in each class are averaged to a final output *o*. The purpose of *TAE* is to extract the commonality among a category. Specifically, for 1-shot setting, there is no average operation. The purpose of *TAE* is to eliminate the differences among instances and extract the commonalities in the same category.

3.2.2. Projector: characteristics among classes

The goal of the second component namely projector is to find the characteristics of various classes. Projector takes the output of *TAE* as input and produce a mask for the support and query set by observing all the support classes at the same time.

$$o : (N, m_2, w_2, h_2) \xrightarrow{reshape} \hat{o} \xrightarrow{CNN} p : (1, m_3, w_3, h_3) \tag{7}$$

During the projector process, firstly, we reshape the $o : (N, m_2, w_2, h_2)$ into $\hat{o} : (1, N \times m_2, w_2, h_2)$, then a small CNN is applied to $\hat{o}$, producing the mask $p : (1, m_3, w_3, h_3)$. Finally, a *softmax* is also applied to the dimension $m3$. For making the output of projector p influence the feature encoder output $f_\varphi(.)$, we need to match the shape between p and $f_\varphi(.)$. This could be achieved as follows:

$$f_\varphi(.) \xrightarrow{Reshaper} r(.) : (N \times K, m_3, w_3, h_3) \tag{8}$$

where *Reshaper* means a light-weight CNN network, and $r(.)$ is regarded as the *Reshaper* network.

3.2.3. Portable to backbone

The task-adaptive module is portable, which could be easily integrated with any metric-based few-shot learning methods. In this paper, we investigate two classical metric-based methods: prototypical network (inductive) and transductive propagation network (transductive) in Sec 2.2, both do not consider the whole support set at the same time. For support set, mask p directly onto the embedding. For the query set, $\odot$ stands for broadcasting the value of p along the sample dimension (NK) in $\mathcal{Q}$. So the distance measurement could be modified as follows. Specifically, θ is a model parameter in TPN.

$$f_{sim}(\mathcal{S}, \mathcal{Q}, \varphi, \theta) = \mathcal{M}(p \odot r(f_\varphi(\mathcal{S})), p \odot r(f_\varphi(\mathcal{Q}))) \tag{9}$$

Our loss function is based on the cross entropy following most of state-of-the-art methods:

$$\mathcal{L} = \frac{exp(\sum_j^K f_{sim}(x_j, x_q))}{\sum_{i=1}^N exp(\sum_j^K f_{sim}(x_j^i, x_q))} \tag{10}$$

where x_j^i , x_q denoting support and query example respectively.

4. Experiments

4.1. Experimental setting

Dataset: in our work, ESC-50 [31] and noiseESC-50 are used. The ESC-50 dataset contains 2,000 5-seconds audio clips that belonged to 50 classes, each having 40 examples. Our model also follows [13] to evaluate the performance under noise condition called noiseESC-50 that selected from audio recordings of 15 different acoustic scenes from the DCASE2016 [1]. So, the performance on ESC-50 and noiseESC-50 reflects the generalization ability of the model in real-world applications. What's more, the dataset of ESC-50 is relatively smaller than AudioSet, which suffers from the class imbalance problem. The iteration times is set as 60 and the initial learning rate as 0.01. We set weight decay to 10^{-4} to avoid overfitting.

Data preparation: To directly compare our model with other baselines, we follow the setting of [18] as same as [13]. Two datasets are divided into 35 classes for training, 10 classes for test and other classes for validation. All audio clips are downsampled from 44.1kHz to 16kHz. 128-bin log mel-spectrogram of raw audio is extracted as the input. The librosa [32] is used for feature extraction. During the episodic training, for each task, we only perform the mixup on the query set $\mathcal{Q}$. Empirically, $\tau = 24$, $\upsilon = 36$ and $M = 2$ are used for time and frequency masking, and $\alpha = 0.2$ is used for mixup.

4.2. Performance on ESC-50 and noiseESC-50

Experimental results are shown in Table 1. Our model significantly outperforms the previous model on two datasets. As shown in Table 1, the absolute improvement of our best model (TA+TPN+DA (temporal&channel)) over published SOTA (TPN) is +6.8% in *5-way 1-shot*, +8.4% in *10-way 1-shot* on ESC-50. On noiseESC-50, +5.9% in *5-way 1-shot* and +7.8% in *10-way 1-shot*. At the same time, on ESC-50, we also notice that the performance gains slightly improvement than the SOTA, +0.1% in *10-way 5-shot*, 0.3% in *5-way 5-shot* on ESC-50, +3.8% in *10-way 5-shot*, +1.0% in *5-way 5-shot* on noiseESC-50. Obviously, our model gains more improvement in 1-shot setting. Two metric-based methods can be continuously improved by integrating TA. Specifically, the experiment results of TPN on two datasets is produced by ourself. All results are averaged over 1000 episodes.

From the statistics of Table 1, data augmentation and temporal&channel attention mechanism are also contributive. The

Table 1: *The result of sound detection (in %) on ESC-50 and noiseESC-50. All baselines reported here are directly reprint the experimental results from the literature [13].* ***TA*** *means the task-adaptive module.* ***DA*** *means the data augmentation method.*

Model	ESC-50				noiseESC-50			
	5-way acc		10-way acc		5-way acc		10-way acc	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MatchingNet [18]	53.7%	67.0%	34.5%	47.9%	51.0%	61.5%	31.7%	43.0%
RelationNet [20]	60.0%	70.3%	41.7%	52.0%	56.2%	74.5%	39.2%	52.5%
SimilarityEmbeddingNet [33]	61.0%	78.1%	45.2%	65.7%	63.2%	78.5%	44.2%	62.0%
ProtoNet [19]	67.9%	83.0%	46.2%	74.2%	66.2%	83.0%	46.5%	72.2%
ProtoNet + AS [13]	74.0%	87.7%	55.0%	76.5%	69.7%	85.7%	51.5%	73.5%
TPN [17]	74.2%	86.9%	55.2%	76.7%	72.7%	86.1%	52.7%	72.9%
ProtoNet+DA	70.5%	83.3%	48.9%	74.7%	69.8%	83.3%	47.7%	72.1%
TA+ProtoNet	70.2%	84.0%	48.6%	74.8%	69.5%	83.5%	50.1%	72.4%
TA+ProtoNet+DA	70.8%	84.6%	50.9%	75.3%	71.5%	84.8%	50.2%	72.3%
TA+ProtoNet+DA (temporal&channel)	71.6%	85.2%	51.5%	75.7%	72.1%	85.2%	51.3%	72.9%
TPN+DA	77.3%	86.5%	60.1%	75.7%	77.1%	86.6%	55.9%	73.1%
TA+TPN	76.5%	87.1%	57.8%	**76.9%**	76.3%	86.3%	57.3%	73.2%
TA+TPN+DA	80.2%	86.4%	62.3%	76.1%	77.9%	86.8%	59.3%	76.1%
TA+TPN+DA (temporal&channel)	**81.0%**	**87.2%**	**63.6%**	76.8%	**78.6%**	**87.1%**	**60.5%**	**76.7%**

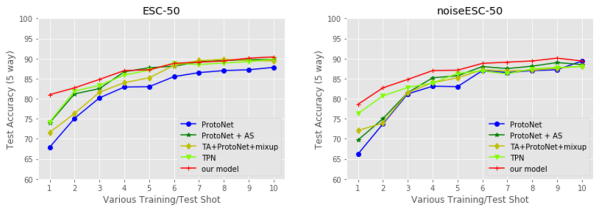

Figure 3: *5-way performance with various training/test shots*

data augmentation strategy could continuously improve the effect of various few-shot models. The enhancement is more obvious of TA+TPN, 2.7% for 5-way 1-shot and 2.0% for 10-way 1-shot. The improvement brought by data augmentation and novel attention mechanism illustrate that the performance of baseline methods is severely underestimated.

4.3. Analysis of experimental results

First, the temporal&channel attention can significantly improve the sound's representation, thus promote model's performances. It is acknowledged that temporal&channel attention and data augmentation strategy could reduce the intra-class variation [34]. With regarding the noise, the performance on ESC-50 is inferior to noiseESC-50. In addition, another significant observation is that 5-shot is less significantly improved than 1-shot. For example, in 5-way of ESC-50, the performance of our model over published state-of-the-art is 0.3% for 5-shot but 6.8% for 1-shot. Moreover, the margin of the various model decreases with the increasing of shots is because more labeled data are used. The superiority of task-adaptive module and other modules will be decreased when more labeled data are available. In this paper, We also make detail experiments (5-way k-shot $k \in \{1, 2, ..., 10\}$) of various model, and the results are presented in Figure 3, which verifies the viewpoint above.

4.4. Analysis of domain mismatch

During this subsection, we follow the setting created by [35]. While the current evaluation focus on recognizing novel class with limited training examples, these novel classes are sampled from the same domain. So, we follow the experiments from [35], such a out-of-domain testing could display the ability of few-shot learning methods to generalize [36, 37]. Following the previous setup, the selected AudioSet [38] with 99 events for meta-train, meta-validation with 21 events and meta-test with 21 events. In addition, the pre-processing are same as the previous description about the setup of [13]. The number of ways is set to 5, and we rerun the experiments: ProtoNet, ProtoNet+AS and TPN. For the domain mismatch setting, we experiments "Music" and "Animals" domain as same as [35]. The events and associated audios from the two domains are removed from train set.

Table 2: *The result of few-shot sound detection in domain mismatch. The AUC(Area Under Curve) is used for evaluation.*

Model (AUC)	1-shot		5-shot	
	Music	Animal	Music	Animal
ProtoNet[19]	0.712	0.644	0.824	0.729
ProtoNet+AS[13]	0.736	0.677	0.839	0.750
TPN[17]	0.747	0.685	0.843	0.748
ProtoNet+DA	0.752	0.680	0.836	0.739
TA+ProtoNet	0.759	0.691	0.851	0.763
TA+ProtoNet+DA	0.762	0.694	0.854	0.761
TA+ProtoNet+DA(temporal&channel)	**0.779**	**0.705**	**0.857**	**0.768**
TPN+DA	0.755	0.693	0.848	0.751
TA+TPN	0.762	0.691	0.854	0.753
TA+TPN + DA	0.768	0.696	0.850	0.754
TA+TPN+DA(temporal&channel)	**0.779**	**0.705**	**0.857**	**0.761**

5. Conclusion

In this paper, we proposed a task-adaptive module for few-shot sound event detection. The module contains a task-adaptive extractor (TAE) and a projector. By considering all support examples at same time, TAE could extract the inner-class commonality and the projector could find cross-class characteristic features. Besides, the data augmentation strategy and the novel attention mechanism could further improve model's performance. We demonstrated that it significantly improved accuracy on two benchmarks (ESC-50 and noiseESC-50), achieving state-of-the-art performance. In addition, we compared with several baselines under the experimental domain mismatch setting, our model could also gain improvement over baselines.

6. References

[1] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *Proceedings of EUSIPCO*, 2016, pp. 1128–1132.

[2] Y. Chen, H. Dinkel, M. Wu, and K. Yu, "Voice activity detection in the wild via weakly supervised sound event detection," in *Proceedings of Interspeech*, H. Meng, B. Xu, and T. F. Zheng, Eds. ISCA, 2020, pp. 3665–3669.

[3] S. G. Upadhyay, B. Su, and C. Lee, "Attentive convolutional recurrent neural network using phoneme-level acoustic representation for rare sound event detection," in *Proceedings of Interspeech*, H. Meng, B. Xu, and T. F. Zheng, Eds. ISCA, 2020, pp. 3102–3106.

[4] W. Xia and K. Koishida, "Sound Event Detection in Multichannel Audio Using Convolutional Time-Frequency-Channel Squeeze and Excitation," in *Proc. Interspeech 2019*, 2019, pp. 3629–3633.

[5] T.-W. Su, J.-Y. Liu, and Y.-H. Yang, "Weakly-supervised audio event detection using event-specific gaussian filters and fully convolutional networks," in *Proceedings of ICASSP*. IEEE, 2017, pp. 791–795.

[6] X. Zheng, Y. Song, J. Yan, L.-R. Dai, I. McLoughlin, and L. Liu, "An Effective Perturbation Based Semi-Supervised Learning Method for Sound Event Detection," in *Proc. Interspeech 2020*, 2020, pp. 841–845.

[7] Y.-H. Shen, K.-X. He, and W.-Q. Zhang, "Learning How to Listen: A Temporal-Frequential Attention Model for Sound Event Detection," in *Proc. Interspeech 2019*, 2019, pp. 2563–2567.

[8] J. Liu and Y. Yang, "Event localization in music auto-tagging," in *Proceedings of ACMMM*, 2016, pp. 1048–1057.

[9] J. Pons, J. Serrà, and X. Serra, "Training neural audio classifiers with few data," in *Proceedings of ICASSP*. IEEE, 2019, pp. 16–20.

[10] H. Song, J. Han, S. Deng, and Z. Du, "Acoustic Scene Classification by Implicitly Identifying Distinct Sound Events," in *Proc. Interspeech 2019*, 2019, pp. 3860–3864.

[11] I. Park and H. K. Kim, "Two-Stage Polyphonic Sound Event Detection Based on Faster R-CNN-LSTM with Multi-Token Connectionist Temporal Classification," in *Proc. Interspeech 2020*, 2020, pp. 856–860.

[12] Y. Wang, J. Salamon, N. J. Bryan, and J. P. Bello, "Few-shot sound event detection," in *Proceedings of ICASSP*. IEEE, 2020, pp. 81–85.

[13] S. Chou, K. Cheng, J. R. Jang, and Y. Yang, "Learning to match transient sound events using attentional similarity for few-shot sound recognition," in *Proceedings of ICASSP*, 2019, pp. 26–30.

[14] W. Wang, C.-C. Kao, and C. Wang, "A simple model for detection of rare sound events," in *Proc. Interspeech 2018*, 2018, pp. 1344–1348.

[15] S. Zhang, Y. Qin, K. Sun, and Y. Lin, "Few-shot audio classification with attentional graph neural networks," in *Proceedings of Interspeech*, G. Kubin and Z. Kacic, Eds. ISCA, 2019, pp. 3649–3653.

[16] V. G. Satorras and J. B. Estrach, "Few-shot learning with graph neural networks," in *Proceedings of ICLR*, 2018.

[17] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. J. Hwang, and Y. Yang, "Learning to propagate labels: Transductive propagation network for few-shot learning," in *Proceedings of ICLR*, 2019.

[18] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proceedings of NIPS*, 2016, pp. 3630–3638.

[19] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *Proceedings of NIPS*, 2017, pp. 4077–4087.

[20] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of CVPR*, 2018, pp. 1199–1208.

[21] W. Chen, Y. Liu, Z. Kira, Y. F. Wang, and J. Huang, "A closer look at few-shot classification," in *Proceedings of ICLR*. OpenReview.net, 2019.

[22] V. G. Satorras and J. B. Estrach, "Few-shot learning with graph neural networks," in *Proceedings of ICLR*. OpenReview.net, 2018.

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

[24] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Proceedings of ICLR*. OpenReview.net, 2017.

[25] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *CoRR*, vol. abs/1803.02999, 2018.

[26] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2, 2015.

[27] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proceedings of ICML*, 1999, pp. 200–209.

[28] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong, "Transductive multi-view zero-shot learning," *IEEE Trans. PAMI.*, vol. 37, no. 11, pp. 2332–2345, 2015.

[29] 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," in *Proceedings of Interspeech*. ISCA, 2019, pp. 2613–2617.

[30] Y. Chen and H. Jin, "Rare Sound Event Detection Using Deep Learning and Data Augmentation," in *Proc. Interspeech 2019*, 2019, pp. 619–623.

[31] K. J. Piczak, "ESC: dataset for environmental sound classification," in *Proceedings of ACMMM*, 2015, pp. 1015–1018.

[32] P. Raguraman, M. Ramasundaram, and M. Vijayan, "Librosa based assessment tool for music information retrieval systems," in *Proceedings of MIPR*, 2019, pp. 109–114.

[33] Y. Huang, S. Chou, and Y. Yang, "Generating music medleys via playing music puzzle games," in *Proceedings of AAAI*, 2018, pp. 2281–2288.

[34] X. Song, Z. Wu, Y. Huang, D. Su, and H. Meng, "Specswap: A simple data augmentation method for end-to-end speech recognition," in *Proceedings of Interspeech*, H. Meng, B. Xu, and T. F. Zheng, Eds. ISCA, 2020, pp. 581–585.

[35] B. Shi, M. Sun, K. C. Puvvada, C. Kao, S. Matsoukas, and C. Wang, "Few-shot acoustic event detection via meta-learning," in *Proceedings of ICASSP*. IEEE, 2020, pp. 76–80.

[36] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. S. Lempitsky, "Domain-adversarial training of neural networks," in *Domain Adaptation in Computer Vision Applications*, ser. Advances in CVPR, G. Csurka, Ed. Springer, 2017, pp. 189–209.

[37] S. Motiian, Q. Jones, S. M. Iranmanesh, and G. Doretto, "Few-shot adversarial domain adaptation," in *Proceedings of NeurIPS*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 6670–6680.

[38] J. F. Gemmeke, D. P. W. 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 *Proceedings of ICASSP*, 2017, pp. 776–780.