FEW-SHOT CONTINUAL LEARNING FOR AUDIO CLASSIFICATION

Yu Wang[♯*] *Nicholas J. Bryan*[♭] *Mark Cartwright*[♯] *Juan Pablo Bello*[♯] *Justin Salamon*[♭]

[♯]Music and Audio Research Laboratory, New York University, NY, USA
[♭]Adobe Research, San Francisco, CA, USA

ABSTRACT

Supervised learning for audio classification typically imposes a fixed class vocabulary, which can be limiting for real-world applications where the target class vocabulary is not known a priori or changes dynamically. In this work, we introduce a few-shot continual learning framework for audio classification, where we can continuously expand a trained base classifier to recognize novel classes based on only few labeled data at inference time. This enables fast and interactive model updates by end-users with minimal human effort. To do so, we leverage the dynamic few-shot learning technique and adapt it to a challenging multi-label audio classification scenario. We incorporate a recent state-of-the-art audio feature extraction model as a backbone and perform a comparative analysis of our approach on two popular audio datasets (ESC-50 and AudioSet). We conduct an in-depth evaluation to illustrate the complexities of the problem and show that, while there is still room for improvement, our method outperforms three baselines on novel class detection while maintaining its performance on base classes.

Index Terms— Continual learning, few-shot learning, supervised learning, audio classification

1. INTRODUCTION

Audio classification is a well-studied research field [1–5] with a wide variety of applications such as multimedia search and retrieval [4], urban sound monitoring [6], bioacoustic monitoring [7], and audio captioning [8]. Most recent audio classification methods employ a standard *supervised learning* approach applied to deep neural networks. While successful, this approach has two significant drawbacks: it requires large quantities of labeled data and can only detect classes that were included in these data, i.e., it imposes a fixed class vocabulary. These requirements, while seemingly innocuous, can make a majority of audio classification methods unusable for applications where the target class vocabulary is not known a priori. That is, many real-world scenarios require us to customize the class vocabulary, such as adding new classes, for example, to personalize the wake-up-words on smart devices, to monitor new bird species at different locations, or to transcribe rare musical instruments.

As an alternative, few-shot learning [9–14] has been applied to audio classification [15–17] and sound event detection [18, 19], where a classifier must learn to recognize a novel class from very few examples. Among different few-shot learning methods, metric-based prototypical networks [12] have been shown to yield excellent performance for audio [15, 18, 19]. However, few-shot methods do not maintain the training data class vocabulary, requiring manual labeling of all novel classes for deployment, which can be overwhelming for large vocabulary problems.

*This work was performed while interning at Adobe Research. This work is partially supported by National Science Foundation award 1544753.

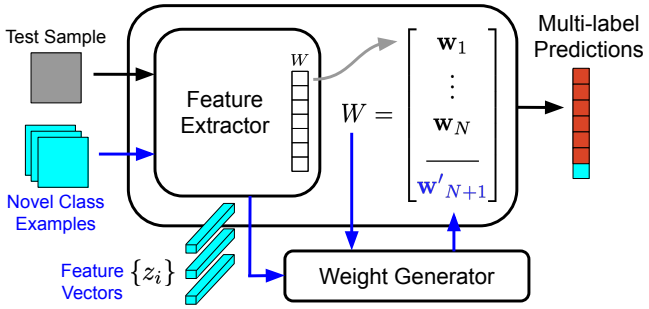

Fig. 1. Few-shot continual learning for multi-label audio classification. A sample (grey) is labeled with one or more base classes (red) defined at train time and novel classes (blue) defined at inference time without retraining, using only few examples per novel class.

Continual learning (never-ending learning, incremental learning, lifelong learning) [20–22], in contrast, is an online learning strategy where an algorithm seeks to continuously adapt to a sequence of tasks and perform well on all tasks without forgetting. It has been proposed for sound classification [23] and sound event detection [24] to learn new sound events without forgetting the previously learned ones. However, continual learning approaches typically require retraining when introducing novel classes, complicated training procedure, or large amounts of labeled data of the novel classes, which are not ideal for practical application with resource-constrained computing environments or audio domains.

Recently, the novel research field of few-shot continual learning (few-shot incremental learning, low-shot learning) combines the strengths of the aforementioned approaches and aims to continuously expand the capability of a classifier based on only few data at inference time [25–28]. This enables fast and interactive model updates by end-users. In this work, we (1) introduce a few-shot continual learning audio classification framework by leveraging the previously proposed dynamic few-shot learning technique (DFSL) [25] as shown in Figure 1. Initially, we train a classifier on *base classes* with abundant examples and extend it at inference time to recognize previously unseen *novel classes* based on few labeled data while not forgetting base classes. We tackle audio specific challenges including multi-label and weak labels by (2) updating the problem definition and loss in DFSL, and (3) incorporating a recent state-of-the-art audio feature extraction model [29] as a backbone. We conduct a comparative analysis of our approach on two popular audio datasets, (4) provide in-depth evaluation of base and novel class performance to elucidate the difficulties of the problem, and (5) show that our method outperforms three baselines on novel class classification while maintaining performance on base classes. To the best of our knowledge, this is the first work introducing few-shot continual learning to the audio domain.

2. METHODS

2.1. Dynamic few-shot learning (DFSL)

Dynamic few-shot learning (DFSL) [25] is a classification approach that aims to learn novel categories from only a few labeled data points while not forgetting the base categories on which it was initially trained. To classify a given audio input, a typical supervised convolutional neural network (CNN) classifier extracts a feature vector $z \in \mathbb{R}^d$ from the audio signal and compute per-class likelihoods by applying a set of classification weight vectors $w \in \mathbb{R}^d$, one per class, to the features. In this context, DFSL extends a typical classifier with an additional module, the few-shot classification weight generator, which can generate a classification weight vector for a novel class. The weight generator takes only K labeled examples of a novel class as input (typically $K \leq 5$), and exploits past knowledge by incorporating an attention mechanism over the existing classification weight vectors of N base classes as shown in Figure 1, where

$$w'_{N+1} = \phi_{avg} \odot z_{avg} + \phi_{att} \odot w'_{att}. \quad (1)$$

The resulting novel classification weight vector w'_{N+1} is a weighted sum of the averaged feature vector z_{avg} and the attention-based weight vector w'_{att}, where ϕ_{avg}, $\phi_{att} \in \mathbb{R}^d$ are learnable weights, $\odot$ is the Hadamard product, $z_{avg} = \frac{1}{K}\sum_{i=1}^{K} z_i$, and w'_{att} is given by

$$w'_{att} = \frac{1}{K}\sum_{i=1}^{K}\sum_{b=1}^{N} Att(\Phi_q z_i, k_b) \cdot w_b, \quad (2)$$

which can be viewed as a linear combination of the base classification weight vectors $W_{base} = \{w_b\}_{b=1}^{N}$. The weighting of each vector is computed via an attention kernel *Att(.,.)*, which is a cosine similarity function followed by a softmax over the base classes. z_i is the feature vector of the i_{th} novel example, $\Phi_q \in \mathbb{R}^{d\times d}$ is a learnable matrix for query vector transformation, and $\{k_b \in \mathbb{R}^d\}_{b=1}^{N}$ is a set of learnable keys for memory indexing. Combining the generated novel weight vector w'_{N+1} with the original base weight vectors W_{base}, we can jointly predict base and novel classes in one unified framework. Note that there can be more than one novel class.

2.2. Training the few-shot classification weight generator

Following past work [25], given a training set with N base classes and a standard classifier pre-trained on these base examples, we train the attention-based few-shot classification weight generator as follows. In each training iteration, we first sample M "pseudo" novel classes from the base classes, simulating the novel classes we see at inference time. For each pseudo-novel class, we sample K training examples to generate its new weight vector via the few-shot weight generator. This results in a new classification weight matrix W^*, which is a union of the generated pseudo-novel weights W'_M and the original weights for the remaining base classes W_{N-M}. We can then update the weight generator parameters and base classification weight vectors to minimize the classification loss on a batch with both base and pseudo-novel classes. Note that the feature extraction model is fixed after pre-training.

2.3. DFSL for multi-label, weakly-labeled audio classification

DFSL was originally proposed for multi-class (single-label) object recognition in images. A critical characteristic of audio data is that multiple sounds can overlap in time, making real-world audio classification a multi-label problem. To adapt DFSL to the audio domain, we replace the (categorical) cross-entropy loss with a binary cross-entropy loss to train a model that can predict multiple concurrent classes. In addition, we use a powerful 14-layer CNN, which achieved state-of-the-art audio tagging results [29], as the feature extraction backbone. A global temporal pooling is applied after the last convolutional layer to summarize the features, which is critical for training with *weakly-labeled* audio data.

3. EXPERIMENTAL DESIGN

To evaluate our proposed few-shot continual learning via DFSL approach for audio classification, we apply it to two audio datasets, ESC-50 and AudioSet, and compare its performance against three baseline methods under a realistic evaluation setup where the test set includes samples of both base classes and novel classes.

3.1. Datasets

ESC-50 consists of 2000 five-second environmental audio recordings [30]. Data are balanced between 50 classes, with 40 examples per class, covering animal sounds, natural soundscapes, human sounds (non-speech), and ambient noises. ESC-50 is small, with a single sound source per clip, minimal background noise, and validated labels, making it high-quality but also simple and unrealistic. Note, while clips in ESC-50 are originally single-labeled, in our experiment, we treat them as multi-labeled to directly compare it with AudioSet by converting each class label to a one-hot vector.

AudioSet is a large-scale collection of over 2M human-labeled ten-second sound clips drawn from YouTube videos [4]. It provides comprehensive coverage of real-world sounds spanning 527 sound classes organized in a hierarchical taxonomy. Despite its large size, AudioSet is a challenging dataset due to its incomplete labeling (missing labels), weak labels (no timing information), and label noise. Each clip in AudioSet can be labeled with multiple sound events, but not all events within a clip are guaranteed to be labeled. A quality assessment showed that a substantial number of classes have poor annotation accuracy[1]. Since label noise is not the focus of this study, to mitigate its impact on our evaluation, we use a subset of accurately-labeled *leaf* classes. To this end, we only use leaf-classes with an annotation quality of 80% or above[1], resulting in 140 classes.

3.2. Baselines

We compare our proposed approach against three baseline methods: *Retrain*, *Prototypical Network*, and *Base classifier + Prototype*. In the first baseline, we simply combine the base training data with the few additional novel class examples to update the training set and *retrain* the classifier. While conceptually simple, this approach requires us to indefinitely store all the training data (original and new) and repeatedly retrain whenever we encounter a new class.

Our second baseline is a pure few-shot learning approach based on a *Prototypical Network* [12]. The model is trained on the base classes to learn a discriminative feature space. Given this space, a prototype representation can be computed for a novel class by averaging the feature vectors of a few novel examples. The distance between these class prototypes and a test datum in feature space encodes similarity. To recognize base classes at inference time, we compute a prototype for each base class by averaging the feature vectors of all the training data available for that class. Note that here too we replace the cross-entropy loss with a binary cross-entropy loss, to support multi-label classification.

[1] https://research.google.com/audioset/dataset/index.html

Our third baseline, introduced in [25], combines a supervised model for the base classes with a simple approach for generating classification weights for novel classes: computing feature vectors for the novel class examples using the supervised model, and treating their average as the weight vector for the novel class: $w'_{N+1} = z_{avg}$. This *Base classifier + Prototype* approach is a naive way to extend a pre-trained classifier to novel classes that relies solely on its feature extraction layers without leveraging its classification weight matrix.

3.3. Training

We partition the classes in each dataset into three splits: *base*, *novel (val)*, and *novel (test)* with a ratio of 30:10:10 for ESC-50, and 100:20:20 for AudioSet. In ESC-50, we split samples in each base class into 24:8:8 for training, validation, and testing. For AudioSet, we use base-class samples from its released *unbalanced train set* for training and validation with a 10:1 ratio. We downsample each audio clip to 16 kHz and compute a 64-bin log-scaled Mel-spectrogram as the input to the model using librosa [31] with a window length of 25 ms, hop size of 10 ms, and a fast Fourier transform size of 64 ms.

For our approach, we first train a supervised base classifier comprising a feature extraction model and a base classification weight matrix. Note that for ESC-50, since it only has 24 training samples per base class, we pre-train the feature extraction model for all methods (except *Prototypical Network*) on AudioSet, and only fine-tune the last linear layer (the weight matrix) on ESC-50. Then we train the attention-based few-shot weight generator following the steps in Section 2.2, where $(M, K) = (5, 5)$, on batches of 100 samples of pseudo-novel classes and 100 samples of the remaining base classes. We implement models in PyTorch [32] and use the Adam optimizer [33] with a learning rate of 0.001 for 60,000 batches with early stopping. A validation batch is similar to a training batch but replacing the pseudo-novel classes with classes in the *novel (val)* split. The baselines are trained using the same input representation and feature extraction architecture. We train the prototypical network using standard *5-way 5-shot* classification tasks [12] sampled from the base training set.

3.4. Evaluation

For evaluation we use actual novel classes as opposed to the pseudo-novel classes used for training. We sample 5 labeled examples from each class in the *novel (test)* split to infer novel classification weight vectors, and combine them with the base weight matrix to extend the classification vocabulary. We evaluate the updated classifier on a test set of samples from both base and novel classes. We run 100 iterations of this process to account for sampling randomness and report the averages for the following metrics: per-class mean average precision (mAP) and F-measure (using 0.5 as the threshold), which we aggregate separately across base and novel classes. mAP summarizes the quality of a model's precision-recall curve, while the F-measure indicates model performance for a fixed threshold.

The ESC-50 test set has 320 examples, 8 examples for each class in the *base* and *novel (test)* splits. Since we use most base class samples for training, we are only left with 8 samples for testing, and so we use the same number of test samples for the novel classes for a balanced evaluation. For the AudioSet test-set we use samples corresponding to the *base* and *novel (test)* classes from AudioSet's released *evaluation set*. While it is maximally-balanced with at least 50 positive examples for as many classes as possible, some classes occur more frequently. We do not discard any samples from this test set to maintain comparability with previous studies.

Note that in the previous work, when DFSL was evaluated on jointly classifying a mixture of samples from both the base and novel classes in a joint label space, only the final metric aggregated across all classes was reported [25]. To add insight, we look at classification performance on base and novel classes separately to show how vocabulary expansion affects base and novel class recognition along with the confusions between them.

4. RESULTS

4.1. ESC-50

In Table 1 we present the classification performance on the test set for our proposed approach and the baselines. For reference we also include the performance of the supervised *Base classifier* that is trained on base classes only.

We see that retraining the supervised model with the few additional novel examples yields the highest mAP on both base and novel classes and an F-measure comparable to other approaches. This may be due to our use of transfer learning (pre-training on AudioSet). Given that ESC-50 is a relatively easy dataset, fine-tuning with few examples appears effective. For real-world applications, however, retraining may not be an option as it requires on-device optimization, dataset storage, and a heavy computational cost, making it either infeasible or impractical for, e.g., mobile devices.

On the other hand, DFSL achieves the highest F-measure on novel classes with only a slight drop in base class performance compared to the original base classifier. That is, it is able to learn to classify novel classes by only seeing a few examples per class, while not forgetting the base classes. Unlike retraining, it does not require storing the base training data or any additional training. DFSL also outperforms the *Base classifier + Prototype* method on novel classes by a large margin, showing that given few novel examples, using the weight vector generated by the few-shot weight generator for classification is notably more effective than using the feature averages.

The prototypical network does not work well under our evaluation scheme. When we take a closer look, we see that it does not output likelihoods over 0.5 and therefore requires threshold tuning for meaningful F-measures. Threshold tuning, however, defeats the purpose of working with only a few data points per novel class. Therefore, we only present the mAP scores for the prototypical network. The feature space learned from scratch on *5-way 5-shot* classification tasks using base classes is not discriminative enough to classify 30 base classes and 10 novel classes when combined. This shows that the standard few-shot learning approach would require further modification for continual learning.

Lastly, as a sanity check, we compute the performance of the base classifier and DFSL model trained and evaluated on ESC-50 under its original single-label, multi-class setup, reported at the bottom of Table 1. As expected, performance increases under this formulation.

4.2. AudioSet

Next, we perform the evaluation on our 140-class subset of AudioSet. Here the feature extraction model for all methods is trained from scratch since we have a large amount of data for all base classes. The results in the upper section of Table 2 show that DFSL outperforms all baselines on novel classes, especially on F-measure, and again with only a slight drop in performance for the base classes. AudioSet is a much more challenging and realistic dataset compared to ESC-50, with a larger vocabulary, label noise, incomplete and

Method	Vocab	mAP		F-measure	
		Base	Novel	Base	Novel
Base clf.	30	0.78	–	**0.62**	–
Retrain	30 + 10	**0.82**	**0.74**	0.59	0.52
Proto. Net	30 + 10	0.17	0.21	–[2]	–[2]
Base clf. + Prototype	30 + 10	0.78	0.64	**0.62**	0.26
DFSL (Ours)	30 + 10	0.78	0.67	0.59	**0.53**
Base clf. (single-labeled)	30	–	–	0.72	–
DFSL (single-labeled)	30 + 10	–	–	0.71	0.65

Table 1. Test performance on ESC-50 on predicting 30 base and 10 novel classes with only 5 labeled examples for each novel class.

Method	Vocab	mAP		F-measure	
		Base	Novel	Base	Novel
Base clf.	100	0.55	–	**0.50**	–
Retrain	100 + 20	0.51	0.15	0.44	0.06
Proto. Net	100 + 20	0.19	0.07	–[2]	–[2]
Base clf. + Prototype	100 + 20	0.55	0.19	**0.50**	0.08
DFSL (Ours)	100 + 20	**0.56**	**0.20**	0.48	**0.21**
Base clf.	50	0.58	–	0.53	–
DFSL (pseudo-novel)	50 + 20	**0.59**	0.22	**0.54**	0.24
DFSL (novel)	50 + 20	**0.59**	**0.24**	0.49	**0.26**

Table 2. Test performance on AudioSet on predicting 100 base and 20 novel classes with only 5 labeled examples for each novel class.

weak labels, and multiple sound sources per clip. We see that retraining or using prototype features with only 5 labeled data points per novel class fails to generalize. With the larger vocabulary, the prototypical network performs even worse on novel classes compared to ESC-50. We see that not only can DFSL achieve continual learning based on just a few examples per novel class at inference time, but it can also work under challenging and practical conditions.

4.3. Error analysis

In Table 2, we see that while DFSL outperforms all baselines on novel classes, there is still a large gap between its performance on base and novel classes. To gain further insight, in Figure 2 we show the DFSL confusion matrices for both ESC-50 and AudioSet. To build a confusion matrix for multi-labeled data, given a test data point, we first count all correctly predicted labels, then we attribute each false prediction evenly across all ground truth labels. We normalize each row by the number of examples with the corresponding label.

We note an asymmetry between novel classes confused as base versus base confused as novel, indicating that novel classes are over predicted in both datasets. With DFSL, we can generate reasonable classification weight vectors for novel classes, but, since they are based on just a few labeled data points, they fail to fully discriminate between acoustically similar classes. For example, in ESC-50, the novel class *Sea waves* often gets activated by samples labeled as *Rain*, *Train*, *Airplane*, or *Washing Machine*. In AudioSet, the novel class *Banjo* often gets activated by samples labeled as *Acoustic Guitar*, which is acoustically similar, but also by samples labeled as other musical instruments such as *Cello*, *Tabla*, and *Accordion*. These instruments do not necessarily sound like a banjo, but they

[2]Requires threshold tuning to compute meaningful F-measures.

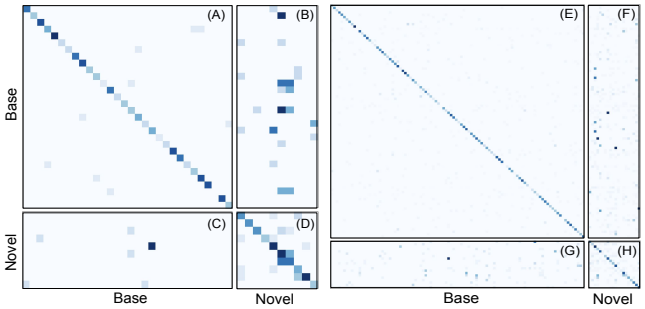

Fig. 2. ESC-50 (lefIt) and AudioSet (right) confusion matrix from DFSL models, where y axes represent ground truth labels and x axes represents model predictions. All axes are split into a set of base classes and a set of novel classes. (A, E) Base confused as base, (B, F) base confused as novel, (C, G) novel confused as base, and (D, H) novel confused as novel. Darker colors represents higher counts.

are likely to exist in similar acoustic scenes. This type of confusion resulting from label co-occurrences is specific to multi-labeled data.

4.4. Training with novel instead of pseudo-novel classes

To bring DFSL performance on novel classes in AudioSet closer to its base class performance, we experiment with several training variations such as using a weighted loss, adding regularization, using a smaller feature extraction model, and varying the number of pseudo-novel classes as well as the number of labeled examples per class for training. However, most of these variations did not lead to significant improvement. The most impactful variation we have identified so far is to better match the train and test scenarios by training the few-shot weight generator on *actual* novel classes, i.e., classes the model has never seen before, as opposed to pseudo-novel classes that are sampled from the base classes the model has already seen. To experiment with this idea, we split the 100 base classes in half, train the base model on the first 50 classes, and train the few-shot weight generator by sampling novel classes from the remaining 50 classes. The results at the bottom of Table 2 show that training with actual novel classes improves performance on novel classes compared to training with pseudo-novel classes, albeit at the cost of slightly lower base class performance. As part of future work we plan to investigate improvements to the attention mechanism of the weight generator as well as entirely replacing the weight generator with a discriminator that directly predicts the novel class likelihood.

5. CONCLUSION

In this paper we propose a few-shot continual learning framework for audio classification, which can expand its base classification vocabulary to novel classes at inference time given just few labeled examples. To this end, we adapt the dynamic few-shot learning technique (DFSL) to multi-label audio classification, which extends a standard base classifier with an attention-based few-shot weight generator. We evaluate the proposed approach on two audio datasets, ESC-50 and AudioSet, comparing its performance with several baselines, and conduct an error analysis to gain further insight. We also propose a training variation to further improve performance on novel classes. While there is still room for improvement, our results show that DFSL is able to achieve few-shot continual learning under challenging and practical conditions without requiring any re-retraining.

6. REFERENCES

[1] H. Zhang, I. McLoughlin, and Y. Song, "Robust sound event recognition using convolutional neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[2] M. Valenti, A. Diment, G. Parascandolo, S. Squartini, and T. Virtanen, "Dcase 2016 acoustic scene classification using convolutional neural networks," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2016.

[3] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "Dcase 2017 challenge setup: Tasks, datasets and baseline system," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2017.

[4] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[5] S. Hershey et al., "Cnn architectures for large-scale audio classification," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017.

[6] J. P. Bello, C. Silva, O. Nov, R. L. Dubois, A. Arora, J. Salamon, C. Mydlarz, and H. Doraiswamy, "Sonyc: A system for monitoring, analyzing, and mitigating urban noise pollution," *Commun. ACM*, pp. 68–77, 2019.

[7] J. Salamon, J. P. Bello, A. Farnsworth, M. Robbins, S. Keen, H. Klinck, and S. Kelling, "Towards the automatic classification of avian flight calls for bioacoustic monitoring," *PloS one*, 2016.

[8] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: an audio captioning dataset," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[9] G. R. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML Workshop*, 2015.

[10] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*, 2017.

[11] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *International Conference on Learning Representations*, 2017.

[12] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Neural Information Processing Systems*. 2017.

[13] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[14] W. Chen, Y. Liu, Z. Kira, Y. F. Wang, and J. Huang, "A closer look at few-shot classification," in *International Conference on Learning Representations*, 2019.

[15] J. Pons, J. Serrà, and X. Serra, "Training neural audio classifiers with few data," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[16] S. Zhang, Y. Qin, K. Sun, and Y. Lin, "Few-shot audio classification with attentional graph neural networks," in *Interspeech*, 2019.

[17] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[18] Y. Wang, J. Salamon, N. J. Bryan, and J. P. Bello, "Few-shot sound event detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[19] Y. Wang, J. Salamon, M. Cartwright, N. J Bryan, and J. P. Bello, "Few-shot drum transcription in polyphonic music," in *International Society for Music Information Retrieval Conference*, 2020.

[20] J. C. Schlimmer and D. Fisher, "A case study of incremental concept induction," in *AAAI*, 1986.

[21] R. S. Sutton, S. D. Whitehead, et al., "Online learning with random representations," in *International Conference on Machine Learning*, 2014.

[22] M. B. Ring, "Child: A first step towards continual learning," in *Learning to learn*. Springer, 1998.

[23] X. Wang, C. Subakan, E. Tzinis, P. Smaragdis, and L. Charlin, "Continual learning of new sound classes using generative replay," in *Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019.

[24] E. Koh, F. Saki, Y. Guo, C. Hung, and E. Visser, "Incremental learning algorithm for sound event detection," *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, 2020.

[25] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[26] Y. Wang, R. Girshick, M. Hebert, and B. Hariharan, "Low-Shot Learning from Imaginary Data," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[27] H. Qi, D. Lowe, and M. Brown, "Low-shot learning with imprinted weights," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[28] X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, and Y. Gong, "Few-shot class-incremental learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[29] 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," *arXiv preprint arXiv:1912.10211*, 2019.

[30] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *International Workshop on Machine Learning for Signal Processing (MLSP)*, 2015.

[31] B. McFee, V. Lostanlen, M. McVicar, A. Metsai, S. Balke, C. Thomé, C. Raffel, et al., "librosa/librosa: 0.7.0," July 2019.

[32] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, et al., "Pytorch: An imperative style, high-performance deep learning library," in *Neural Information Processing Systems*. 2019.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.