WHO CALLS THE SHOTS? RETHINKING FEW-SHOT LEARNING FOR AUDIO

Yu Wang,[1*] *Nicholas J. Bryan*[2]*, Justin Salamon*[2]*, Mark Cartwright,*[3] *Juan Pablo Bello*[1]

[1] Music and Audio Research Laboratory, New York University, New York, NY, USA,
[2] Adobe Research, San Francisco, CA, USA
[3] New Jersey Institute of Technology, Newark, NJ, USA

ABSTRACT

Few-shot learning aims to train models that can recognize novel classes given just a handful of labeled examples, known as the support set. While the field has seen notable advances in recent years, they have often focused on multi-class image classification. Audio, in contrast, is often multi-label due to overlapping sounds, resulting in unique properties such as polyphony and signal-to-noise ratios (SNR). This leads to unanswered questions concerning the impact such audio properties may have on few-shot learning system design, performance, and human-computer interaction, as it is typically up to the user to collect and provide inference-time support set examples. We address these questions through a series of experiments designed to elucidate the answers to these questions. We introduce two novel datasets, FSD-MIX-CLIPS and FSD-MIX-SED, whose programmatic generation allows us to explore these questions systematically. Our experiments lead to audio-specific insights on few-shot learning, some of which are at odds with recent findings in the image domain: there is no best one-size-fits-all model, method, and support set selection criterion. Rather, it depends on the expected application scenario. Our code and data are available at `https://github.com/wangyu/rethink-audio-fsl`.

***Index Terms*—** Few-shot learning, continual learning, audio classification, supervised learning, classification

1. INTRODUCTION

The field of few-shot learning, i.e., training a model such that it can recognize previously unseen classes given a small set of examples, has seen significant progress in recent years, especially in the image domain [1–6]. Advances have also been made in the audio domain [7–13], including for few-shot continual learning where a model can recognize a predefined set of base classes *and* learn to recognize new classes given few examples [14], but many challenges still remain for this domain.

Audio data have unique characteristics that set them apart from image data. Most importantly, sound events may overlap, leading to attributes such as polyphony (i.e., how many sounds are overlapped), per-sound-event signal-to-noise (foreground to background) ratios, and weakly labeled data. While the computer vision literature has focused mostly on strongly labeled, single-label images (i.e., multi-class problems) [1–4, 6], many audio recognition problems like sound event detection (SED), instrument recognition, and speaker recognition are often multi-label.

The multi-label nature of many audio recognition problems complicates the inference-time support set example selection process for users of few-shot learning systems. It is no longer merely a question of how many examples to provide (i.e., the size of the *support set*). Now, users must determine how to compose the best support set in terms of each sample's polyphony and SNR. For example, is it okay to provide examples with other overlapping sounds? Should the support set only contain a given target sound? How does the background noise level of the support set affect performance?

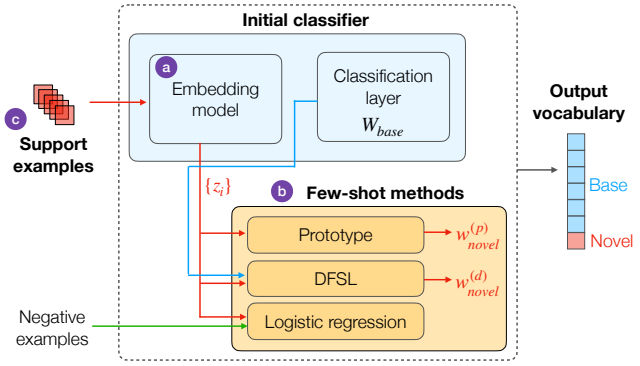

Figure 1: Few-shot continual learning framework for multi-label audio classification. We analyze (a) embedding model architectures, (b) few-shot methods, and (c) support set selection.

In this paper, we elucidate the relationship between few-shot model design (architecture), optimization strategy (few-shot method), and inference-time support set example selection within the framework of few-shot continual learning for multi-label audio classification, as shown in Figure 1. Our analysis leads us to insights that are unique to the audio domain, sometimes in contradiction with recent findings in computer vision [15]. Our contributions can be summarized as follows:

- We introduce novel datasets, FSD-MIX-CLIPS and FSD-MIX-SED, allowing us to study the impact of audio-specific qualities (polyphony, SNR) on few-shot learning in a controlled way.
- We show that there are design choices between few-shot models, methods, and inference-time support set selection and that there is currently no one-size-fits-all solution. Rather, it depends on the anticipated application scenario that includes desired user labeling effort, runtime computation and storage resources, and prior knowledge of the test data.
- We provide audio-specific insights for few-shot learning including guidance on the choice of support set in terms of size, polyphony, and sound event SNR that can be used for both system design and motivate new few-shot learning methods.

*This work is partially supported by NSF award 1544753.

2. THE FSD-MIX-CLIPS AND FSD-MIX-SED DATASETS

We begin by surveying past audio datasets, including ESC-50 [16], AudioSet [17] and FSD50K [18]. ESC-50 contains a small set of environmental recordings of 50 classes. It is high-quality, but also too simple and unrealistic for our needs as it contains single-source audio and minimal background noise. In contrast, AudioSet is a large-scale dataset of over 2M human-labeled 10s sound clips drawn from YouTube videos with 527 sound classes. AudioSet, however, is not fully open for audio download and the annotation quality varies notably between classes[1]. Such issues motivated the development of FSD50K [18], which is a fully open dataset that contains over 51k audio clips manually labeled using 200 classes drawn from the AudioSet Ontology and content gathered from Freesound [19]. Unfortunately, FSD50K does not provide strong labels of events across time and polyphony, and does not contain SNR annotations – both central to our study. Thus, we build upon this work to construct a programmatically-mixed dataset, FSD-MIX-CLIPS, in which we control the polyphony and SNR characteristics. Table 1 shows the number of classes and mixed clips in each split in FSD-MIX-CLIPS, as well as the number of single-labeled clips from FSD50K used as source material for mixing. In a few-shot continual learning framework, we need disjoint sets of *base* and *novel* classes where novel class data are only used at inference time.

To generate clips with multiple sound events, we first filter FSD50K to get a set of single-labeled clips as the foreground sound material. We choose clips shorter than 4s that have a single validated label with the *Present and Predominant* annotation type [18, 20]. We further trim the silence at the edges of each clip using pysox [21] with a threshold of 0.1% and a minimum duration of 0.01s. The resulting subset contains clips each with a single and strong label. The 200 sound classes in FSD50K are hierarchically organized. We focus on the leaf nodes and rule out classes with less than 20 single-labeled clips. This gives us 89 sound classes and a total of around 10k single-labeled clips. Next, we partition the 89 classes into three splits: *Base*, *Novel-val*, and *Novel-test* with 59, 15, and 15 classes, respectively. For base classes, we follow the original data split in FSD50K, where clips in the development set are used for training and validation with a ratio of 5:1, and clips in the evaluation set are used for testing. Novel classes are only used at inference time during validation or testing. Thus, for each novel class, we combine the clips from the development and evaluation sets of FSD50K.

We generate 10-second strongly-labeled soundscapes with controlled SNR and polyphony with Scaper [22]. We use the single-labeled FSD50K clips as foreground sounds and Brownian noise as the background. We define the number of classes c in each soundscape by drawing from 1 to 5 with probability proportional to $1/c$. Then we sample c classes from one of the class splits without replacement. From each of the c classes, we sample one clip as a foreground sound event. Each sound event is randomly pitch-shifted within +/- 2 semitones and time-stretched by a ratio in [0.8, 1.2]. All transformed clips are then randomly placed in a soundscape and mixed with an SNR randomly sampled from [-5, 20] dB. We refer to this (intermediate) dataset of 10s soundscapes as FSD-MIX-SED.

Finally, we extract 615k 1s clips centered on each sound event in the soundscapes in FSD-MIX-SED to produce FSD-MIX-CLIPS. To label a clip, we consider all sound events within the 1s window. If an event overlaps with the window for more than 0.5s or half of the event duration, we add the corresponding class into the clip label. We then consider the number of classes within a clip as the level of polyphony with the assumption that it is rare to have short non-overlapping events within a 1s window.

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

Class split	**Base**			**Novel-val**	**Novel-test**
# Classes	59			15	15
Data split	**Train**	**Val.**	**Test**	**Val.**	**Test**
FSD-MIX-SED (10s)	200k	30k	30k	8k	8k
FSD-MIX-CLIPS (1s)	450k	65k	65k	17k	17k
FSD50K clips used	$\sim$5k	$\sim$1k	$\sim$2k	$\sim$1k	$\sim$1k

Table 1: Numbers of classes and mixed clips/soundscapes in FSD-MIX-CLIPS (1s) and FSD-MIX-SED (10s) per split and the corresponding number of single-labeled clips used from FSD50K.

One of our goals is to study the impact of support set polyphony and SNR, and it is cleaner to do so under a multi-label audio classification setting (i.e., tagging), as opposed to SED where post-processing plays an important factor. For this reason, the remainder of this work will focus on FSD-MIX-CLIPS. By making both datasets available, we hope to encourage researchers to build on this work, e.g., by translating this study to a full SED setting.

3. EXPERIMENTAL DESIGN

We perform a series of experiments with FSD-MIX-CLIPS to understand design choices between few-shot model architectures, learning methods, and support set selection under the few-shot continual learning framework for multi-label audio classification. Moreover, to clearly elucidate support set selection issues, we specifically analyze different support set compositions, including the number of support examples, polyphony, and SNR.

3.1. Embedding model architectures

We first investigate two common convolutional neural network (CNN) model architectures, *Conv4* and *PANN*, to understand how embedding model architecture and capacity affect overall few-shot learning performance. *Conv4* consists of four convolution blocks, each of which has a convolutional layer with a 3×3 kernel, a batch normalization layer, a ReLU activation layer, and a 2×2 max-pooling layer. The first two convolutional layers have 64 feature channels and the latter two have 128 feature channels. We flatten the last feature map to get output embeddings with dimension 3072 and $\approx$ 440k trainable parameters. On the other hand, *PANN* is a powerful 14-layer CNN [23], which achieved state-of-the-art audio tagging results. The output embedding has 2048 dimensions, and the number of trainable parameters is $\approx$80M.

Besides training the embedding model from scratch, we also investigate using a pre-trained OpenL3 embedding model [24, 25]. Here, we seek to understand if pre-training through self-supervised learning of audio-visual correspondence in YouTube videos provides a more generalizable representation. We extract OpenL3 embeddings with dimension 512 from 1s audio inputs, and train an additional fully-connected layer to transform the embedding dimension to 2048 with $\approx$1M trainable parameters. We refer to this model as *pre-OpenL3+FC*. To disentangle the effect of pre-training and model capacity, we also train an embedding model with the *OpenL3+FC* architecture from scratch without pre-training.

3.2. Few-shot methods

We investigate three few-shot methods, *Prototype*, *DFSL*, and *LR*, to understand how different learning paradigms affect performance and their relationship to the support set. To predict a novel class based on few data, both the *Prototype* and *DFSL* approaches aim to compute a new classification weight vector to add to the existing base weight matrix as shown in Figure 1(b), while *LR* aims to learn a new binary logistic regression model directly.

The *Prototype* approach [4, 26] treats the averaged embedding z of the support examples for a novel class as a prototype, and uses it as a novel classification weight vector: $w^{(p)}_{novel} = z_{avg} = \frac{1}{n}\sum_{i=1}^{n} z_i$, where n is the number of support examples. This approach is a simple way to extend the initial classifier to novel classes that relies solely on the embedding model. Dynamic Few-Shot Learning (*DFSL*) [10, 26] goes a step further to train a few-shot classification weight generator to generate novel classification weight vectors. The weight generator not only takes in the prototype of a novel class z_{avg} but also exploits past knowledge of the initial classifier by incorporating a cosine-similarity based attention mechanism over the base classification weight matrix W_{base} to get $w^{(d)}_{novel} = \phi_{avg} \odot z_{avg} + \phi_{att} \odot w'_{att}$. Here, w'_{att} can be viewed as a learned weighted sum of base class weight vectors in W_{base}. ϕ_{avg} and ϕ_{att} are learnable weights, and $\odot$ is the Hadamard product. The few-shot weight generator is trained via episodic training [27], where each training iteration mimics the testing scenario. First, 5 "pseudo" novel classes are sampled from the base classes. We treat each pseudo novel class as if it is an actual novel class at inference time, and sample n examples from it. Then we generate new weight vectors for these pseudo-novel classes based on support examples and W_{base}. We replace the classification weight vectors of pseudo-novel classes temporally and update the few-shot weight generator parameters to minimize the classification loss on a batch of data with both base and pseudo-novel classes.

Lastly, recent few-shot image classification work has shown that a simple logistic regression or *LR* model trained on top of a pre-trained embedding outperforms many meta-learning based algorithms [15]. It suggests that good learned embeddings are capable of fine-tuning on few data. We evaluate this approach to see if it is as effective for multi-label audio classification under a few-shot continual learning setup. At inference time, for each novel class, we train a binary LR model on the support example embeddings. Note that because the support examples are labeled examples of a novel class, we also need negative examples to train a binary classifier. Since the training data does not include any of the novel classes, we randomly sample x examples from the training set as negative examples, where x is a hyperparameter tuned on the validation set.

3.3. Support set selection

Support examples play an important role in the few-shot continual learning paradigm, as they are the only supervision given for each novel class. So, we vary the number of support examples n from 5, 10, 20, to 30, representing standard few-shot and low data scenarios. Then, given n, we experiment with audio-specific characteristics, polyphony and SNR, of the support examples. First, we experiment with *monophonic* and *polyphonic* support sets. *Monophonic* means we only include examples containing a single sound event from the target novel class. *Polyphonic* means the examples may contain, in addition to the target class, other overlapping sounds. Next, we experiment with two versions, *low-SNR* and *high-SNR*, of monophonic support sets. We define the SNR values within [-5, 0, 5] (dB) as *low-SNR*, and [10, 15, 20] (dB) as *high-SNR*.

Note that for few-shot methods that involve model training with support examples at either train time or inference time (*DFSL* and *LR*), the support set characteristics are matched between training and testing in our experiments.

3.4. Training and evaluation setups

Given our FSD-MIX-CLIPS dataset, embedding model architectures, few-shot learning methods, and a set of distinct support set characteristics, we start by training the initial classifier on the *Base* training set. 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 [28] with a 25 ms window length, 10 ms hop size, and fast Fourier transform size of 64 ms. We implement the classifier in PyTorch [29] and use the Adam optimizer [30] with a 0.001 learning rate with early stopping. Then, if *DFSL* is used as the few-shot method, we further train the few-shot classification weight generator in an additional training stage, also on the *Base* training set.

At evaluation, for each class in the *Novel-test* split, we sample a set of support examples. By combining the remainder of the novel class data with the test data for base classes, we get a test set which is a mixture of base and novel class samples. If *LR* is used as the few-shot method, we train one binary logistic regression model for each novel class using the support examples. We use scikit-learn [31] implementation for the logistic regression model with balanced class weights and a maximum of 1000 iterations. Finally, we combine the initial base classifier and the few-shot method to predict the test set labels in a joint label space. We run 100 iterations of this evaluation procedure to account for sampling randomness, and compute averaged per-class F-measure with a fixed threshold of 0.5. We aggregate the per-class F-measure and compute the mean over base and novel classes separately. In addition to computing metrics based on class labels, the automatic annotation produced by Scaper for FSD-MIX-CLIPS allows us to break down model performance by test-set polyphony, ranging from 1 to 4, and by SNR, ranging from -5 to 20 dB.

4. RESULTS

4.1. Embedding model architectures

In Figure 2, we show the performance on base and novel classes with different embedding models. All other variables are kept fixed in this set of experiments, with *DFSL* as the few-shot method, monophonic support examples with mixed SNR, and $n = 5$.

First, we see that the base class performance increases with increasing model capacity. This matches our intuition that standard multi-label classification benefits from a more powerful model. Second, this trend does not hold for novel classes. *PANN* achieves the highest base class performance but low novel class performance, suggesting a trade-off between overfitting base classes and generalizing to novel classes. Third, *pre-OpenL3+FC* achieves the best novel class performance and the most balanced performance between base and novel classes. This can potentially be attributed to pre-training as *OpenL3+FC*, which is trained from scratch, shows lower novel class performance. We conjecture pre-training helps prevent overfitting to base classes and improves generalization to novel classes–this model has "seen" significantly more data during pre-training via self-supervision on unlabeled data. We fix the embedding model to *pre-OpenL3+FC* in the rest of the work.

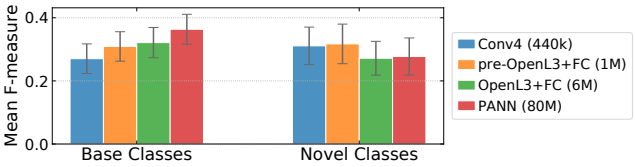

Figure 2: Mean F-measures averaged over 59 base classes and 15 novel classes. The number of learnable parameters are shown after model names. Error bars represent 95% confidence intervals.

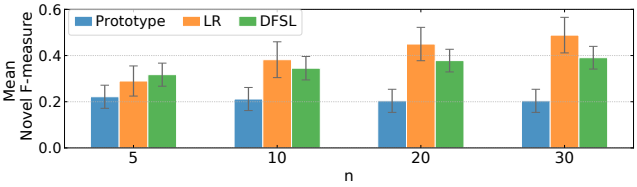

Figure 3: Mean F-measures averaged over 15 novel classes. Error bars represent 95% confidence intervals.

4.2. Few-shot methods and number of support examples

Next, we look at the performance of different few-shot methods on predicting novel classes with n varied from 5 to 30, representing standard few-shot to low data scenarios. Here, the support sets are monophonic with mixed SNR. The results, shown in Figure 3, provide several insights. First, we find that *DFSL* performs the best out of all methods when $n = 5$, a typical few-shot learning scenario. Among the three methods we experimented with, *DFSL* was specifically designed to solve few-shot learning problems as it leverages an additional episodic training stage to train the few-shot weight generator. Interestingly, this finding is in contrast to the conclusions reached for single-label, multi-class, few-shot image classification [15]. Second, as n increases to 10 or more, *LR* starts to outperform *DFSL*. This indicates that a simple transfer learning approach becomes more effective when the number of support examples increases. The downside of *LR*, however, is that it requires additional optimization and data storage at inference time, since we need to train a binary logistic regression model for each novel class. Training such binary models, as mentioned in Section 3.2, requires additional negative examples randomly sampled from the training data. The optimal number of negative examples at different n, based on validation performance, ranges from 100 to 5000. We need to store the embeddings of these negative examples to train each binary model. On the other hand, while *DFSL* requires an additional training stage, it only requires a forward pass at inference time to generate novel classification weight vectors to predict novel classes.

We can formulate these findings into the following design choices. When there is no constraint on annotation budget or computation and storage resources, *LR* as the few-shot method with large n is the best combination. However, if it is critical to minimize user labeling or runtime resources, *DFSL* with small n is preferable.

4.3. Polyphony and SNR of support and test samples

Lastly, we show how the polyphony and SNR of the support set affect model performance when predicting novel classes in test samples. Here, we focus on the $n = 5$ few-shot scenario for both *LR* and *DFSL*, but we found similar trends for $n = 30$.

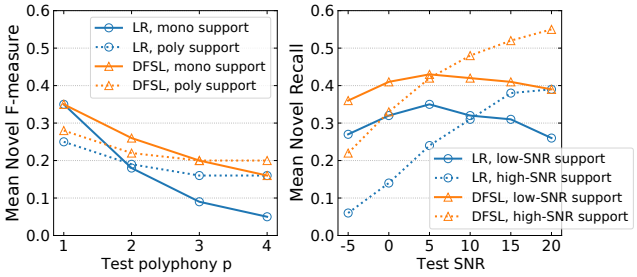

Figure 4: Performance breakdown by (left) polyphony p and (right) SNR of test samples. Note that we compute recall in the right plot since we can only define SNR when the sound is present.

First, the results, displayed in Figure 4, show that matching the support set characteristics to those of the test samples, both in terms of polyphony and SNR, gives the best results. For both few-shot methods, the best performance on monophonic and highly polyphonic test samples is achieved by monophonic and polyphonic support sets, respectively. Similarly, the best performance on test samples with low/high SNR is achieved by support sets with low/high SNR, respectively. Second, Figure 4 (left) shows that polyphonic support sets lead to more consistent performance across test set polyphonies, while the highest overall performance comes from monophonic support sets on monophonic test samples. Figure 4 (right) shows that support sets with low SNR lead to more consistent test performance across SNR, while support sets with high SNR achieve the highest recall on high-SNR test samples. Lastly, we see that *LR* exhibits a dramatic performance drop when there is a mismatch between the polyphony of the support and test samples. A possible explanation is that each binary LR model is trained directly on the support examples, and therefore, the model performance is more dependent on support example characteristics. Whereas for *DFSL*, the base weight matrix is included as additional information.

We can distill these observations into two reproducible insights for our multi-label few-shot continual learning audio classification task. First, if we know the test sample characteristics (e.g., mostly monophonic with high SNR), matching those in the support set will lead to better performance. Conversely, if we do not have prior knowledge about the test data, opting for support examples with more complex acoustic characteristics (i.e., mixed polyphony and lower SNR) would be a safer choice which leads to more consistent performance. This holds for both the polyphony and SNR characteristics, using either the *DFSL* or *LR* learning methods.

5. CONCLUSIONS

In this work, we study few-shot learning for audio. We create a programmatically-mixed dataset, FSD-MIX-CLIPS, to help us to control important common audio data characteristics including polyphony and SNR. Then, we perform a series of experiments to elucidate how model architecture, learning strategy, and support set selection affect performance of few-shot continual learning on multi-label audio classification. Generalizing to real-audio datasets is a potential future work when datasets with comparable size and detailed annotation become available. Our results lead us to useful audio-specific insights, some of which are at odds with recent findings in the image domain: there is no current one-size-fits-all best performing approach, but rather, design choices should depend on the expected application scenario.

6. REFERENCES

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

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

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

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

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

[12] B. Shi, M. Sun, K. C. Puvvada, C.-C. Kao, S. Matsoukas, and C. Wang, "Few-shot acoustic event detection via meta learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[13] K. Shimada, Y. Koyama, and A. Inoue, "Metric learning with background noise class for few-shot detection of rare sound events," in *IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP)*, 2020.

[14] Y. Wang, N. J. Bryan, M. Cartwright, J. P. Bello, and J. Salamon, "Few-shot continual learning for audio classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[15] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, "Rethinking few-shot image classification: a good embedding is all you need?" *arXiv preprint arXiv:2003.11539*, 2020.

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

[17] 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.

[18] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50k: an open dataset of human-labeled sound events," *arXiv preprint arXiv:2010.00475*, 2020.

[19] F. Font, G. Roma, and X. Serra, "Freesound technical demo," in *ACM international conference on Multimedia*, 2013.

[20] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *ACM International Conference on Multimedia*, 2014.

[21] R. Bittner, E. Humphrey, and J. Bello, "Pysox: Leveraging the audio signal processing power of sox in python," in *International Society for Music Information Retrieval Conference Late Breaking and Demo Papers*, 2016.

[22] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2017, pp. 344–348.

[23] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020.

[24] R. Arandjelovic and A. Zisserman, "Look, listen and learn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[25] J. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3852–3856.

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

[27] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems 29*, 2016.

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

[29] 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.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, 2011.