PSLA: Improving Audio Tagging With Pretraining, Sampling, Labeling, and Aggregation

Yuan Gong, *Member, IEEE*, Yu-An Chung, *Member, IEEE*, and James Glass, *Fellow, IEEE*

***Abstract*—Audio tagging is an active research area and has a wide range of applications. Since the release of AudioSet, great progress has been made in advancing model performance, which mostly comes from the development of novel model architectures and attention modules. However, we find that appropriate training techniques are equally important for building audio tagging models with AudioSet, but have not received the attention they deserve. To fill the gap, in this work, we present PSLA, a collection of model agnostic training techniques that can noticeably boost the model accuracy including ImageNet pretraining, balanced sampling, data augmentation, label enhancement, model aggregation. While many of these techniques have been previously explored, we conduct a thorough investigation on their design choices and combine them together. By training an EfficientNet with pretraining, balanced sampling, data augmentation, and model aggregation, we obtain a single model (with 13.6 M parameters) and an ensemble model that achieve mean average precision (mAP) scores of 0.444 and 0.474 on AudioSet, respectively, outperforming the previous best system of 0.439 with 81 M parameters. In addition, our model also achieves a new state-of-the-art mAP of 0.567 on FSD50K. We also investigate the impact of label enhancement on the model performance.**

***Index Terms*—Audio tagging, audio event classification, transfer learning, imbalanced learning, noisy label, ensemble.**

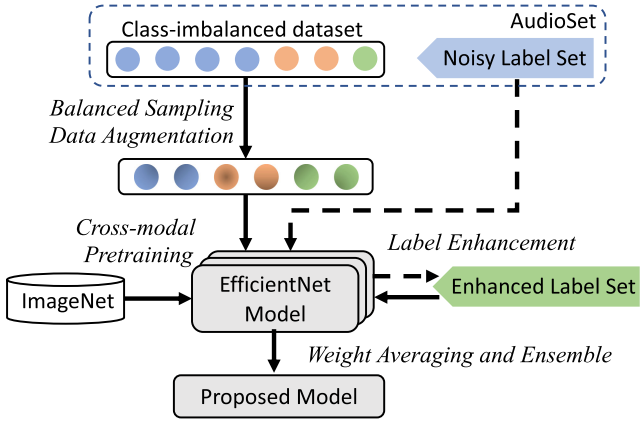

Fig. 1. The proposed Pretraining, Sampling, Labeling, and Aggregation (PSLA) training pipeline. AudioSet is extremely class imbalanced and has prevalent annotation errors, we propose a data augmentation/balanced sampling strategy and a label enhancement strategy to alleviate these two problems. We also pretrain the convolutional neural networks with ImageNet and find it leads to a noticeable performance improvement. By further aggregating multiple models with weight averaging and ensemble techniques, we get a model that performs much better than that trained with a conventional pipeline and achieves a new state-of-the-art mAP of 0.474.

I. Introduction

AUDIO tagging aims to identify sound events that occur in a given audio recording, and enables a variety of Artificial Intelligence-based systems to disambiguate sounds and understand the acoustic environment. Audio tagging has a wide range of health and safety applications in the home, office, industry, transportation, and has become an active research topic in the field of acoustic signal processing.

In recent years, audio tagging and classification research has moved from small and/or constrained datasets such as ESC-50 [1] and CHiME-Home [2] to much larger datasets with a greater variety and range of real-world audio events and substantially more training data. A significant milestone in this field occurred with the release of the AudioSet corpus [3] containing over 2 million 10-second audio clips extracted from video and tagged at the utterance level with a set of 527 event labels. AudioSet is currently the largest and most comprehensive publicly available dataset for audio tagging. Not surprisingly, it has subsequently become the primary source of training and evaluation material for audio tagging research. The availability of AudioSet has encouraged much audio tagging research that has steadily seen the standard evaluation metric of mean average precision (mAP) increase from, for example, 0.314 with shallow fully-connected networks [3], to 0.392 with a residual network with attention [4] to, most recently, 0.439 with spectrogram and waveform-based convolutional neural networks (CNNs) [5]. In order to cope with the weakly labeled data, multiple instance learning and attention mechanisms have also been the subject of much investigation [6]–[9].

In our audio tagging experiments using Audioset we have observed that, in addition to the particular model architecture being evaluated, significant performance improvements can be achieved via training techniques including cross-modal pretraining, data augmentation, label enhancement, and ensemble modeling. Our empirical evaluations show that these model agnostic techniques lead to significant accuracy improvements, and combining them together can further boost the model accuracy. Specifically, we train an ensemble of EfficientNet [10] models with the proposed set of training techniques and achieve

Manuscript received February 1, 2021; revised May 26, 2021 and August 4, 2021; accepted September 2, 2021. Date of publication October 15, 2021; date of current version November 4, 2021. This work was supported in part by Signify. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Wenwu Wang. *(Corresponding author: Yuan Gong.)*

The authors are with Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: yuangong@mit.edu; andyyuan@mit.edu; glass@mit.edu).

Digital Object Identifier 10.1109/TASLP.2021.3120633

a new state-of-the-art mAP of 0.474 on AudioSet, our single model with 13.6 M parameters also achieves an mAP of 0.444, outperforming the previous the best system that contained 81 M parameters. In addition, our model also achieves a new state-of-the-art mAP of 0.567 on the FSD50K benchmark [11].

As shown in Fig. 1, the training techniques we investigated fall into four main categories. First, we find cross-modal pretraining with ImageNet [12] improves the performance of audio tagging CNNs even though AudioSet already contains a substantial amount of in-domain data. Second, we address the Audioset label imbalance by adopting balanced sampling and data augmentation. Third, we observed that there are pervasive annotation errors in AudioSet and studied the impact of such annotation errors on the model performance. We further developed a method to improve training label quality. Fourth, we use weight averaging and ensemble methods to improve the overall performance. Many of these techniques have been proposed previously in isolation. For example, ImageNet pretraining has been used in [13] for small datasets, balanced sampling and data augmentation have been used in [5], label enhancement has been proposed in [14], and ensemble modeling has been used in [4], [15], [16]. To the best of our knowledge however, none of the prior efforts have used more than two of these simultaneously, and the particular implementation is often only briefly mentioned in the literature. In this paper, we thoroughly investigate each of these techniques, a more thorough understanding of the benefits of different training techniques should facilitate a more meaningful comparison between various works because performance differences due to the particular training procedure could overshadow the model architecture or other novel techniques being investigated. The training pipeline we propose is model-agnostic and can serve as a recipe for AudioSet tagging experiments to facilitate fair comparisons with new techniques.

The contributions of this work are summarized as follows:

1) We present a collection of training strategies and design choices for audio tagging. We quantify the improvement of each component via extensive experimentation.
2) By training an ensemble of standard EfficientNet models with the proposed training procedure, we achieve a new state-of-the-art mAP of 0.474 on AudioSet, outperforming the best previous system of 0.439.
3) We release the code, model, and enhanced label set.[1] The training pipeline can serve as a recipe of AudioSet training to facilitate future audio tagging research.

The paper is organized as follows. We first describe the baseline model architecture in Section II, then we gradually improve the baseline model performance on AudioSet by adding new training techniques in Sections III, IV, V, and VI. In each section, we first review the corresponding technique and then present our implementation and results. We present an ablation study, experiments on FSD50K and other model architectures, and a discussion of the results in Section VII. We conclude the paper in Section VIII.

[1]Code at [Online]. Available: https://github.com/YuanGongND/psla

TABLE I
THE AUDIOSET [3] STATISTICS

	Balanced Train	Full Train	Evaluation
AudioSet	22,176	2,065,161	20,383
Downloaded	20,785	1,953,082	19,185
Downloaded Ratio	93.7%	94.6%	94.1%

II. EXPERIMENT SETTING AND BASELINE MODEL

A. Dataset

In this work, we mainly focus on AudioSet [3], a collection of over 2 million 10-second audio clips excised from YouTube videos and labeled with the sounds that the clip contains from a set of 527 labels. AudioSet is a weakly labeled and multi-label dataset, i.e., labels are given to a clip with no indication of where in the clip the associated sound occurred, and every clip can, and often does, have multiple labels associated with it. As shown in Table I, the dataset is split into three subsets: balanced train, unbalanced train, and evaluation. In this paper, we combine the balanced and unbalanced training set as the full training set. The balanced train dataset is a set of 22,176 recordings, where each class has at least 49 samples, while the full train set contains the entire 2 million recordings. The evaluation set consists of 20,383 recordings and contains at least 59 examples for each class. To obtain the raw audio, we extracted the dataset from YouTube. Due to the constant change in video availability (e.g., videos being removed, taken down) there is a natural shrinkage (about 5%) from the original dataset [3]. Specifically, we downloaded 20,785 (94%), 1,953,082 (95%), and 19,185 (94%) recordings for the balanced train, full train, and evaluation set, respectively, which is consistent with previous literature (e.g., [5]). Therefore, we do make fair comparisons with previous state-of-the-art models by evaluating on the same subset of the evaluation dataset.

We also evaluate the proposed PSLA training framework on FSD50K [11], a recently collected data set of sound event audio clips with 200 classes drawn from the AudioSet ontology to see how the PSLA framework generalizes. FSD50K contains 37,134 audio clips for training, 4,170 audio clips for validation, and 10,231 audio clips for evaluation. The audio clips are of variable length from 0.3 to 30s with an average of 7.6s (7.1s for the training and validation set, 9.8s for the evaluation set). For both AudioSet and FSD50K, we sample the audio at 16 kHz.

B. Training and Evaluation Details

For all AudioSet experiments in this paper, we train the neural network model with a batch size of 100, the Adam optimizer [17], and use binary cross-entropy (BCE) loss. We use a fixed initial learning rate of 1e-3 and 1e-4 and cut it in half every 5 epochs after the 35^{th} and 10^{th} epoch for all balanced set and full set experiments, respectively. The reason why a smaller learning rate is used for the full AudioSet is that the full set is about 100 times larger than the balanced set, using a smaller learning rate can avoid the model falling into a local minima before it sees all samples. We use a linear learning rate warm-up

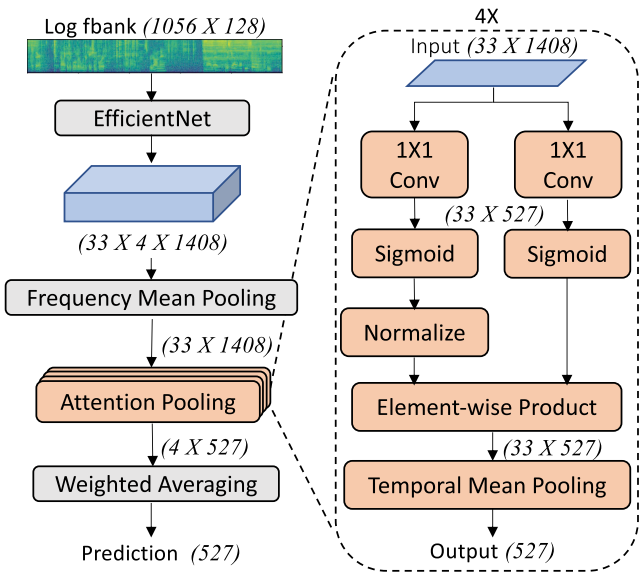

Fig. 2. The audio tagging model used in this work. The 10-second waveform is first converted to a 1056×128 log Mel filterbank (fbank) feature vector and input to the EfficientNet model. The output of the penultimate layer of EfficientNet is a $33 \times 4 \times 1408$ tensor. We apply a frequency mean pooling to produce a 33×1408 representation that is fed into a 4-headed attention pooling module. In each head, the CNN output is transformed into a 33×527 dimensional tensor via a set of 1×1 convolution layers with a parallel attention branch and classification branch. We multiply the output of each branch element-wise and apply a temporal mean pooling (implemented by summation). Finally, we sum the weighted output of each attention head after it has been scaled by a learnable weight and produce the final prediction for all classes.

strategy for the first 1,000 iterations. As in previous efforts, we train the model with 60 and 30 epochs for all balanced set and full set experiments, respectively, and report the mean result on the evaluation set of the last 5 epochs.

We use the mean average precision (mAP) of all the classes as our main evaluation metric since it is the most commonly used audio tagging evaluation metric on AudioSet. Mean average precision is an approximation of the area under a class's precision-recall curve, which is more informative of performance when dealing with imbalanced datasets such as AudioSet and FSD50k compared with the average area under the curve of the receiver operating characteristic curve [18], [19]. In the discussion section, we also report the average area under the curve (AUC) of the receiver operating characteristic curve and sensitivity index (d-prime) in order to compare our model with previous work that only reports AUC and d-prime.

C. Baseline Model

In this work, we use a similar model structure as in [4], illustrated in Fig. 2. Each 10-second audio waveform is first converted to a sequence of 128 dimensional log Mel filterbank (fbank) features computed with a 25ms Hanning window every 10ms. We conduct zero padding to make all audio clips have 1056 frames. This results in a 1056×128 feature vector that is input to a CNN model. In [4] the CNN was based on the ResNet50 model [20]. In our work, the CNN is based on the EfficientNet-B2 model [10] since it requires a smaller

TABLE II
MEAN AVERAGE PRECISION (MAP) COMPARISON OF THE RESNET MODEL [4] AND THE EFFICIENTNET MODEL USED IN THIS PAPER

	# Parameters	Balanced Set	Full Set
ResNet-50	25.66M	0.1635	0.3790
EfficientNet-B2	13.64M	0.1570	0.3723

number of parameters and is faster for training and inference. The EfficientNet model effectively downsamples the time and frequency dimensions by a factor of 32. The penultimate output of the model is a $33 \times 4 \times 1408$ tensor. We apply mean pooling over the 4 frequency dimensions to produce a 33×1408 representation that is fed into a multi-head attention module. The attention module consists of an attention branch and a classification branch. Each branch transforms the CNN mean pooled output into a 33×527 dimensional tensor via a set of 1×1 convolutional filters. After a sigmoid non-linearity and a normalization on the attention branch, we combine the two branches via a element-wise product. A temporal mean pooling (implemented by summation) is then performed to produce a final 527 dimensional output for each class label. Unlike [4], we use a 4-headed attention module instead of a single-head one in this work. We sum the weighted output of each attention head after it has been scaled by a learnable weight to produce the final output.

EfficientNet [10] is a recent proposed convolutional neural network architecture that has shown an advantage on both accuracy and efficiency over previous architectures. Such advantage mainly comes from two design: First, EfficientNet is based on the mobile inverted bottleneck convolution (MBConv) block [21], [22], an efficient residual convolution block. Second, EfficientNet scales the network on all dimensions (i.e., width, depth, and input resolution), which is demonstrated to be a better strategy than scaling only one dimension. In this work, we use EfficientNet-B2 that consists of 9 stages, 339 layers. The original EfficientNet-B2 model for image classification has 9.11M parameters, after adding the attention module and adjusting the classification layer, our audio tagging model has 13.64M parameters in total. As shown in Table II, the EfficientNet model achieves slightly worse performance than the ResNet-50 model, but has 12 million fewer parameters. In the rest of the paper, we keep using the EfficientNet model and show that a significant improvement can be achieved without modifying its model architecture.

III. NETWORK PRETRAINING

Transfer learning and network pretraining have been widely used in computer vision, natural language processing, speech and audio processing in recent years [23]–[25]. The typical process is to first train a model with either a large out-of-domain or unlabeled dataset using an auxiliary task and then fine-tune the model with in-domain data for the main task. The idea being that the knowledge learned from the pretraining task can be transferred to the main task.

TABLE III
PERFORMANCE IMPACT ON MAP DUE TO PRETRAINING WITH IMAGENET DATA

	Balanced Set	Full Set
No pretraining	0.1570	0.3723
With pretraining	0.2385	0.3939

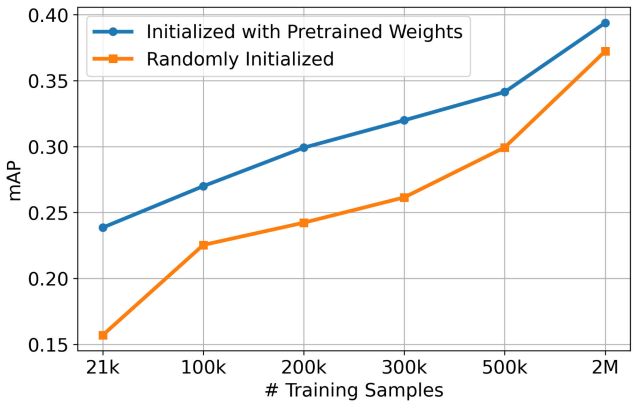

Fig. 3. Comparison of the performance of ImageNet-pretrained model and random-initialized model with different training data volume.

For the audio tagging task, both supervised pretraining (e.g., in [5]) and self-supervised pretraining (e.g., in [26]–[30]) using audio data have been studied in recent years. Performance improvement is typically achieved when the in-domain dataset is small (e.g., ESC-50 [1], UrbanSound [31], and balanced AudioSet). However, it has not been reported that a pretrained model can outperform a state-of-the-art audio tagging model trained from scratch using the full AudioSet, possibly because the full AudioSet contains 2 million audio recordings and there is no larger annotated dataset available. While theoretically self-supervised pretraining can leverage an unlimited amount of unlabelled audio data, in practice it takes effort to find and process large scale data with sufficient variety and coverage of the 527 sound classes.

In contrast to the above-mentioned efforts, we find noticeable performance improvement can be achieved by pretraining the CNN with the ImageNet dataset [12] used for visual object classification, even when the training data for the end task of audio tagging is the full AudioSet. In our experiment, we initialize the EfficientNet (the second to the penultimate layer) with 1) ImageNet-pretrained weights (released by the authors of [10]), and 2) random weights (He Uniform initialization [32]). We then train both models in exactly the same way as described in Section II-B.

As shown in Table III, ImageNet pretraining leads to a 51.9% and 5.8% relative improvement for the balanced set and full set experiment, respectively. To see the relationship between the performance improvement and the end-task training data volume, we further evaluate the performance when the audio tagging training data volume is 100k, 200k, 300k, and 500k (all comprised of the entire balanced set and samples randomly taken from the full set). As shown in Fig. 3, the performance improvement decreases with the training data volume, but is always noticeable. In addition, we find the performance improvement led by ImageNet pretraining is much larger than that led by more training iterations, e.g., when trained with the balanced AudioSet, the model trained with 120 epochs achieves an mAP of 0.1694, which is only slightly better than the model trained with 60 epochs and is significantly worse than the model trained with ImageNet pretraining that achieves an mAP of 0.2385.

In some sense, it is surprising that pretraining a model with data from a different modality can be effective. However, transfer learning from computer vision tasks to audio tasks is not new and has been previously studied in [13], [33]–[35]. However, we believe this is the first time it has been demonstrated to be effective when the dataset of the audio task is at this scale, indicating the auxiliary image classification task helps the model learn some complementary knowledge. We hypothesize that the improvements may be due to the model learning to recognize low-level features such as edges during pretraining. Such knowledge could potentially be relevant for finding acoustic "edges" in the spectrogram.

In practice, many commonly used CNN architectures (e.g., Inception [36], ResNet [20], EfficientNet [10]) have off-the-shelf ImageNet-pretrained models for both TensorFlow and PyTorch. It is also straightforward to adapt these off-the-shelf models to audio tasks. The only things that need to be modified are the first convolution layer and the last classification layer. Since the input of vision tasks is a 3-channel image while the input to the audio task is a single-channel spectrogram, we adjust the input channel of the first convolutional layer from 3 to 1 and initialize it with random weights. Since the classification task is essentially different, we abandon the last classification layer of the pretrained model and feed the output of the penultimate layer to our succeeding layers. We implement this using the `efficientnet_pytorch`[2] package.

In summary, the advantages of using ImageNet pretraining are as follows. First, no additional in-domain labeled or unlabeled datasets are needed. This is important because currently there is no audio tagging dataset of comparable size to AudioSet. Second, ImageNet pretraining can lead to consistent performance improvement even when the in-domain training data size is huge. Third, ImageNet pretraining is practically easy to implement. The limitation is that it is only applicable to models that take 2D image-like input (e.g., spectrogram). Nevertheless, a majority of deep learning models for audio tasks do fall in this category. In the following sections, we use Imagenet pretraining by default for all experiments.

IV. BALANCED SAMPLING AND DATA AUGMENTATION

A. *Balanced Sampling*

As might be expected, the frequency of occurrence of different sound events ranges widely. It is not surprising then that a large scale audio tagging dataset is class imbalanced. As shown in Fig. 4, the most frequent AudioSet class is "Music" which has 949,029 samples, while the most infrequent class "Toothbrush" only has 61 samples, leading to a ratio of 15,557. Such

[2][Online]. Available: https://github.com/lukemelas/EfficientNet-PyTorch

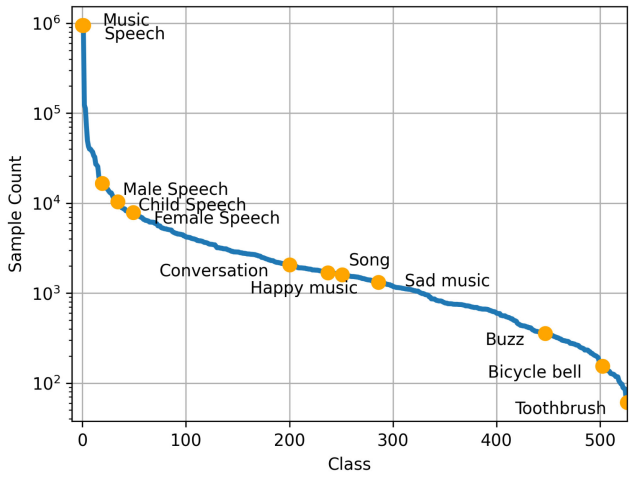

Fig. 4. Sample count of each class in the full AudioSet (vertical axis is in log scale). Note that the sample count of the "Speech" class is substantially larger than the sum of sample counts of the "Male Speech," "Female Speech," and "Child Speech" class. Similarly, the sample count of the "Music" class is substantially larger than the sum of sample counts of the "Happy music" and "Sad music" class. This indicates a potential prevalent miss annotation issue in AudioSet.

imbalances can have a large impact on performance, particularly for low-frequency classes [37].

With such large data imbalance, simple upsampling or downsampling are difficult to implement because upsampling will make the dataset unacceptably large while downsampling will waste a large portion of the data. Moreover, AudioSet is a multi-label dataset, making it even harder to implement up/downsampling methods. In this work, we propose a random balanced sampling method to alleviate the class imbalance problem. Note that balanced sampling on AudioSet has been used in [5], [6], [8], but is only briefly mentioned and the details can only be found in the source code.

The proposed random balanced sampling approach is shown in Algorithm 1, lines 1-8. We first count the sample number c_k of each class k over the entire dataset. We then assign a sampling weight for each sample, specifically, the weight $\boldsymbol{w}^{(i)}$ of the i^{th} sample is $\sum_{k=1}^{527} \mathbb{1}_{\{k \in \boldsymbol{y}^{(i)}\}} 1/c_k$. This assigns a higher weight for samples containing rare audio events and also takes all audio events that appear in the sample into consideration. During training, we still feed N samples (N is the dataset size) to the model for each epoch, but instead of traversing the dataset, we draw a sample from the multinomial distribution parameterized by the above-mentioned sample weights with replacement. That makes rare sound event samples more likely to be seen by the model. The advantages of the proposed random sampling are 1) it is a compromise of upsampling and downsampling. It wastes fewer samples than downsampling while keeping the number of N samples fed to the model every epoch; 2) it is applicable to multi-label datasets; and 3) the model sees a different set of data every epoch, so the model checkpoints after every epoch have a greater diversity, which is helpful for ensembles [38], [39], as we will discuss in Section VI.

As shown in Fig. 5, while the proposed balanced sampling algorithm greatly alleviates the data imbalance issue, the sampled

Algorithm 1 Balanced Sampling and Data Augmentation

Require:
Multi-label Dataset $\mathcal{D} = \{\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\}, i \in \{1, \ldots, N\}$

Procedure 1: Generate Sampling Weight

Input: Label Set $\{\boldsymbol{y}^{(i)}\}$
Output: Sample Weight Set $\mathcal{W} = \{\boldsymbol{w}^{(i)}\}$, $i \in \{1, \ldots, N\}$
1: traverse $\{\boldsymbol{y}^{(i)}\}$, count sample number $\boldsymbol{c_k}$ of each class k
2: initialize $\boldsymbol{w}^{(i)} = 0, i \in \{1, \ldots, N\}$
3: **for** each sample i **do**
4: **for** each class $k \in \boldsymbol{y}^{(i)}$ **do**
5: $\boldsymbol{w}^{(i)} = \boldsymbol{w}^{(i)} + 1/c_k$
return $\mathcal{W} = \{\boldsymbol{w}^{(i)}\}$

Procedure 2: Sampling and Augmentation in Training

Input: $\{\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\}, \mathcal{W}, F, T, M$
6: **for** every epoch **do**
7: **for** $n \in \{1, \ldots, N\}$ **do**
8: draw $i \sim multinomial(\mathcal{W})$
9: **if** $unif(0, 1) <$ mixup rate M **then**
10: draw $j \sim unif\{1, N\}$
11: draw $\lambda \sim Beta(\alpha, \alpha)$
12: $x = \lambda x^{(i)} + (1 - \lambda)x^{(j)}$
13: $y = \lambda y^{(i)} + (1 - \lambda)y^{(j)}$
14: **else**
15: $x = x^{(i)}, y = y^{(i)}$
16: draw $f \sim unif(0, F)$, $f_0 \sim unif(0, 128 - f)$
17: draw $t \sim unif(0, T)$, $t_0 \sim unif(0, 1056 - t)$
18: $x = Masking(f_0, t_0, f, t)(x)$
19: use (x, y) to train the neural network

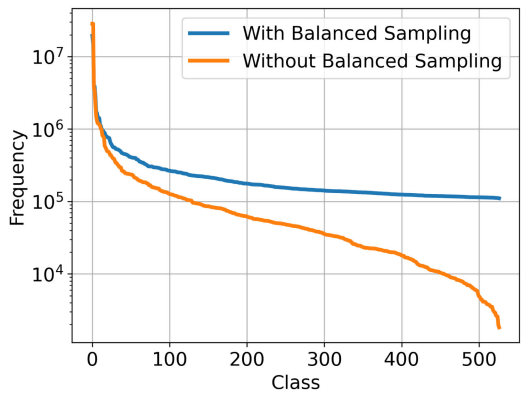

Fig. 5. Sorted sampled frequency of each class after 30 training epochs.

frequency of each class is still imbalanced after the balanced sampling algorithm is applied. This is because AudioSet is a multi-label dataset and minority classes are usually paired with majority classes, thus oversampling the minority class also directly oversamples the majority class. We compare the performance of the model trained with plain dataset traversal (with

TABLE IV
PERFORMANCE IMPACT ON MAP DUE TO VARIOUS BALANCED SAMPLING AND DATA AUGMENTATION STRATEGIES

	Balanced Set	Full Set
Baseline	0.2385	0.3939
+ Balanced Sampling	-	0.3721
+ Time-Frequency Masking	0.2818	0.4265
+ Mix-up Training	0.3108	0.4397

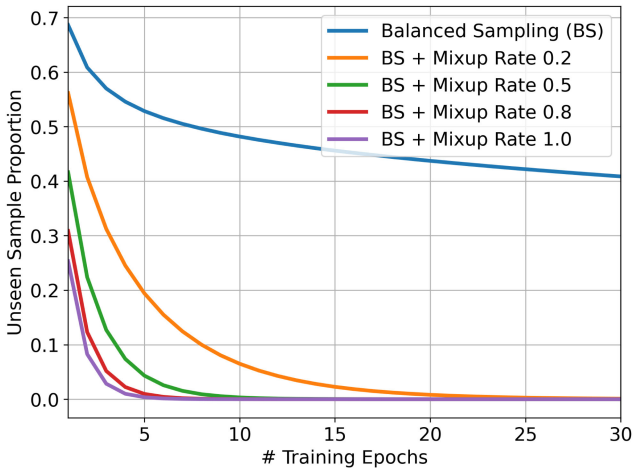

Fig. 6. The proportion of unseen samples with the training epochs. Mixup rate is the probability that the sample input to the model is a mixed-up sample. In our implementation, one of the two mixed-up samples is drawn from a uniform distribution, while the other is drawn using the balanced sampling multinomial distribution.

data reshuffled at every epoch) and with the proposed random sampling. As shown in Table IV, we find random balanced sampling actually lowers the performance. This result is not surprising because: 1) while better than downsampling, there is still a substantial amount of data wasted every epoch. As shown in Fig. 6, 40.9% data is not seen by the model after 30 training epochs; 2) while the low-frequency class samples and high-frequency class samples are roughly equally seen by the model, the low-frequency class samples are actually repeated samples. Both issues increase the risk of model overfitting. Therefore, we explored the use of data augmentation to overcome this problem.

B. Time and Frequency Masking

We first consider simple time and frequency masking for data augmentation, which has been found to be effective for audio tagging [5] and speech recognition [40]. Frequency masking is applied so that f consecutive frequency channels $[f_0, f_0 + f)$ are masked, where $f \sim unif(0, F)$, $f_0 \sim unif(0, 128 - f)$, and F is the maximum possible length of the frequency mask. Similarly, time masking is applied so that t consecutive frequency channels $[t_0, t_0 + t)$ are masked, where $t \sim unif(0, T)$, $t_0 \sim unif(0, 1056 - t)$, and T is the maximum possible length of the frequency mask. Note that 128 and 1056 are the input dimensions of our model. We use the implementation of `torchaudio.transforms.FrequencyMasking` and `TimeMasking`, $F = 48$ and $T = 192$. The masking parameters f_0, t_0, f, t are sampled on-the-fly for each audio sample during training to minimize the chance of repeated audio samples being fed to the model. As shown in Table IV, time and frequency masking improves audio tagging performance considerably, with relative improvements of 18.2% and 14.6% achieved for the balanced set and full set experiment, respectively. Note that the overall amount of training samples per epoch remains the same. We hypothesize that the effectiveness of masking is due to the reduction of repeated samples in the training data, especially for low-frequency samples.

C. Mix-Up Training

An additional form of data augmentation we explored is called *mix-up training* where weighted combinations of audio samples are combined to make new samples. Mix-up training creates convex combinations of pairs of examples and their corresponding labels. Studies have shown it can improve the performance of image classification, voice command recognition [41], [42], and audio tagging [5], [43]. Specifically, mix-up training constructs augmented training examples as follows:

$$x = \lambda x^{(i)} + (1 - \lambda) x^{(j)}$$

$$y = \lambda y^{(i)} + (1 - \lambda) y^{(j)}$$

where $x^{(i)}$ and $x^{(j)}$ are two different training audio samples, $y^{(i)}$ and $y^{(j)}$ are the corresponding labels, $\lambda \in [0, 1]$ and x is the mixed-up new audio sample, and y is the resulting label. We conduct mix-up on the waveform level.

Past explanations for why mix-up training improves performance include: 1) it increases the variation of the training data [5], [43]; 2) it leads to an enlargement of Fisher's criterion in the feature space and a regularization of the positional relationship among the feature distributions of the classes [41], [43]; and 3) it reduces the model's memorization of corrupt labels [42].

In addition to these observations, we find mix-up training has an additional advantage for imbalanced datasets. As we discussed in Section IV-A, balanced sampling, while making the low-frequency class samples more prevalent, has the unfortunate side effect of wasting a large number of (40.9%) class samples. By adopting the mixup strategy, the model can see twice the number of samples within the same training epoch. This advantage can be increased if one of the two mixed-up samples is drawn from a uniform distribution, while the other is drawn using the balanced sampling multinomial distribution introduced in the previous section. Intuitively, mixing up a rare sound event (e.g., toothbrush) with a frequent one (e.g., music) is more reasonable than mixing up two rare sound events. Some previous synthetic audio event detection datasets use a similar method to construct samples [44]. As shown in Fig. 6, the mix-up strategy can reduce the unseen samples to almost zero in just a few epochs.

We further make two modifications based on previous efforts. In prior work λ is drawn from a uniform distribution $unif(0, 1)$ [43] or Beta distribution $Beta(\alpha, \alpha)$ with $\alpha <$

TABLE V
PERFORMANCE AS A FUNCTION OF MIX-UP RATE (TRAINING ON BALANCED SET WITH $\alpha = 10$)

Mixup Rate	0	0.2	0.5	0.8	1.0
mAP	0.2818	0.3060	0.3108	0.3119	0.2928

TABLE VI
PERFORMANCE AS A FUNCTION OF α (TRAINING ON BALANCED SET WITH MIX-UP RATE $= 0.5$)

α	$-\infty$	0.1	1	10
mAP	0.2818	0.3004	0.3087	0.3108

1 [42], where

$$Beta(\alpha, \alpha): \ prob(x; \alpha, \alpha) = \frac{x^{\alpha-1}(1-x)^{\alpha-1}}{B(\alpha, \alpha)}$$

where B is the beta function

$$B(\alpha, \alpha) = \int_0^1 t^{\alpha-1}(1-t)^{\alpha-1} dt$$

Thus λ has a relatively high likelihood to be close to either 0 or 1. From the perspective of sound mixing and reducing the number of unseen samples, a λ close to 0.5 could be more reasonable since it leads to more "evenly" mixed up samples and the model can see both samples. Second, since samples in the evaluation set are not mixed up, mixing up all samples during training might lead to a gap between training and evaluation. Thus we set a mix-up *rate* to control the number of samples to mix up during training, a mixup rate of 0.5 means that 50% training samples are mixup samples and the rest 50% training samples are non-synthetic samples. Therefore, the model can see non-synthetic samples during training. As shown in Fig. 6, a mix-up rate of 0.5 results in 95% samples being seen by the model in 5 epochs. For non mix-up samples, the data loader only needs to load one audio sample instead of two. A low mix-up rate can also reduce the data loading and pre-processing cost during training, which is non-negligible because it is almost impossible to fit the full AudioSet into memory.

We evaluate the impact of mix-up rate and α, as shown in Tables V and VI. A larger α and a medium mix-up rate indeed lead to better classification performance. Combining them achieves 0.3108 mAP, which is better than a plain setting of α=mixup rate=1 that achieves 0.3079 mAP. We use $\alpha = 10$ and mix-up rate$= 0.5$ in all subsequent experiments.

D. Summary

We combine the balanced sampling and masking and mix-up data augmentation strategies together, as described in Algorithm 1. We summarize the contribution of each component in Table IV. It is worth mentioning that while balanced sampling alone lowers the performance, it is helpful when combined with data augmentation strategies. By adopting balanced sampling and data augmentation, an 11.6% relative improvement and an mAP of 0.4397 are achieved for the full set experiment. We only do data augmentation for balanced set experiments as the data is already roughly balanced and obtain a 30.3% relative improvement and an mAP of 0.3108, demonstrating the effectiveness of data augmentation for small datasets. Finally, it is worth mentioning that by merely adopting ImageNet pretraining, balanced sampling, and data augmentation with a standard EfficientNet architecture, the model already outperforms the previous best system. In the following sections, we use balanced sampling (for the full AudioSet) and data augmentation as defaults for all experiments.

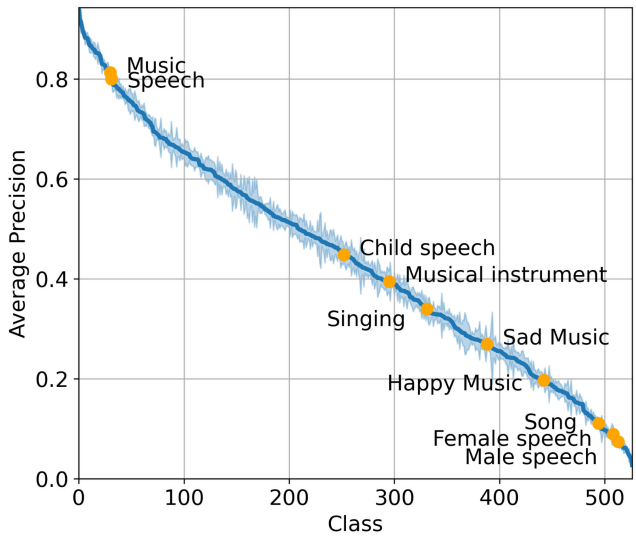

Fig. 7. Sorted class-wise average precision (AP) and its standard deviation of the model trained on full set. Note that the "Speech" class has a much higher AP than the "Male Speech," "Female Speech," and "Child Speech" class. Similarly, the "Music" class has a much higher AP than the "Happy Music" and "Sad Music" class. "Singing" and "Song" have similar definition but very different AP. Classes with low AP also have a larger AP variance.

V. LABEL ENHANCEMENT

In this section, we explore the noisy label aspect of AudioSet: how it impacts audio tagging performance, and how to alleviate it. This line of research is motivated by observing the model's class-wise performance. In Fig. 7, we show the class-wise average precision (AP) of the model trained with the full set. From the figure it is immediately apparent that the AP of each class differs greatly, indicating that the model has a range of ability to recognize various sounds. This is not an issue specific to our model or training pipeline, but has been widely reported in prior work [5], [8], [14], [45], [46]. The order of class-specific performance reported by independent research also appears to be similar. For example, the "Male speech," "Bicycle," "Harmonic," "Rattle," and "Scrape" classes are among the 10 worst performing classes in [45], and they are also are among the 10 worst performing classes for our model when trained with the balanced set. We further confirm that models with different architectures have similar class-specific performance order with experiments in Section VII-B. This consistency suggests that the issue might be due to an intrinsic problem with the data or the task. Since the class-wise AP is not strongly correlated with either class sample count in the training set or the class

TABLE VII
CORRELATION COEFFICIENTS BETWEEN CLASS-WISE AP AND CLASS SAMPLE COUNT/ANNOTATION QUALITY ESTIMATE RELEASED BY AUDIOSET AUTHORS

	Balanced Set	Full Set
AP and Sample Count	0.1692	0.0946
AP and Annotation Quality Estimate	0.2464	0.2629

annotation quality estimate released by the AudioSet authors (as shown in Table VII), it has been hypothesized that the class-wise performance variation is due in part to the difficulty in reliably tagging the different sound classes themselves [5], [46].

While we agree that the poor performance of some classes could be due to particular audio events being difficult to identify, it is not true for all poor-performing classes. For example, the "Male Speech," "Female Speech," and "Child Speech" classes have APs of 0.07, 0.09, 0.45, respectively while the AP of the "Speech" class is 0.80. This discrepancy cannot be explained by the class difficulty hypothesis because recognizing speaker gender from speech is a relatively easy task [47]–[49], and the performances of the speech classes should not be so disparate. By examining the class sample counts, we find another issue that the sample count of the "Speech" class is substantially larger than the sum of sample counts of the "Male Speech," "Female Speech," and "Child Speech" classes. Specifically, in the balanced set, there are 5,309 audio clips with the label "Speech" but only 55, 55, 128 audio clips are with label "Male Speech," "Female Speech," and "Child Speech," respectively. The same thing happens in the full set (shown in Fig. 4): the "Speech Class" has 947,009 samples while the sum of the other three classes is 34,878. In other words, only 4.5% and 3.7% of speech samples are labeled as either male, female, or child speech in the balanced and full AudioSet, respectively. This indicates that a large portion of samples are not correctly labeled. Based on these two observations, we hypothesize that the low performance of the male, female, and child speech classes is not due a small number of samples, or inherent classification difficulty, but that they have only a small fraction of correctly labeled data, which ultimately confuses the model. We refer to this phenomenon as a Type I error.

We also find that there are substantial samples labeled with sub-classes, but not with the corresponding parent class defined by the AudioSet ontology. For example, there are 40 and 3,201 audio clips labeled as either "Male Speech," "Female Speech," or "Child Speech," but not labeled as "Speech" in the balanced and full AudioSet, respectively. We refer to this phenomenon as Type II error.

We formalize the two types of error as follows:

1) Type I error: an audio clip is labeled with a parent class, but not also labeled as a child class when it does in fact contain the audio event of the child class.
2) Type II error: an audio clip is labeled with a child class, but not labeled with corresponding parent classes.

It is worth mentioning that neither type of error are included in the quality estimate released by the AudioSet authors because the quality estimate checked 10 random audio clips of each class and verified that they actually contained the corresponding sound event. In other words, the quality estimate counts the false positive annotation errors, but not false negatives. As a consequence, the quality estimate of the "Male Speech," "Female Speech," and "Child Speech" is 90%, 100%, and 100%, respectively, while they have obvious false negative annotation errors.

Unfortunately, false negatives are prevalent in AudioSet. Another example are the music classes (see Figs. 4 and 7 for sample counts and class-wise AP of music classes). The reason for these types of errors is due to the AudioSet annotation pipeline. In the pipeline, the human annotator verifies the candidate labels nominated by a series of automatic methods (e.g., by using metadata). Also, the list of candidate labels is limited to ten labels per clip. Since the automatic methods for nomination are not perfect, some existing sound events fail to be nominated, or are nominated but ranked below the top ten, thus leading to missing labels [3], [14].

As seen in the speech class example, annotation error can impact performance, but has not received much attention. To the best of our knowledge, only a few efforts have covered the missing label issue. In [45], [50], a synthetic error is studied, however, the real-world noisy labels are believed to be much harder to deal with than the synthetic labels. In [14], the authors propose a loss masking based teacher-student model. In this section, we propose an ontology-based label enhancement method to alleviate the noisy label problem. Our approach differs from previous work in three aspects: First, we work on real-world noisy labels rather than synthetic corrupted labels; Second, we explicitly modify the labels of the training data rather than using loss masking during training. Thus the enhanced label set can be used in the exact same way as the original set (no need to modify the model and training pipeline). We plan to release the enhanced label set to facilitate future research. Third, we leverage the AudioSet ontology to constrain label modification, which reduces the chance of incorrect modifications. For example, for an audio clip labeled as "Speech," we only consider adding child or parent labels in the specific "Speech" branch of the ontology.

As shown in Algorithm 2, the proposed approach consists of the following steps. First, we train a teacher model using the full AudioSet with the original label set. Second, we set a label modification threshold for each audio tagging, specifically, we set the threshold of a class as the teacher model's mean prediction score of all audio clips originally labeled as that class (lines 1-2). The threshold can also be set as other values such as the 5th, 10th, or 25th percentile of the teacher model's prediction score. The lower the threshold, the more labels are added. We then identify all samples that need to be relabeled. For each sample, we compile all child (Type I) and/or parent (Type II) labels of all original labels as the candidate set according to the AudioSet ontology (line 6). For each label in the candidate set, if the teacher model's prediction score of the class is greater than the corresponding label modification threshold, we add it to the labels of the sample (line 7-8). Finally, we retrain the model from scratch with the enhanced label set.

We apply the proposed label enhancement method (with the teacher model's mean prediction score as the label modification threshold) on the balanced training set and show the results in Table VIII. Note the model without label enhancement

Algorithm 2 Label Enhancement

Require:
Teacher Model $\mathcal{M}$
Dataset $\mathcal{D} = \{\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\}, i \in \{1, \ldots, N\}$
Label Ontology $\mathcal{O}$

Procedure 1: Generate Label Modification Threshold

Input: $\mathcal{M}, \mathcal{D}$
Output: Threshold Set $\mathcal{T} = \{\boldsymbol{t}_k\}, k \in \{1, \ldots, 527\}$
for $k \in \{1, \ldots, 527\}$ **do**
  $t_k = \sum_{i=1}^{N} \mathbb{1}_{\{k \in \boldsymbol{y}^{(i)}\}} \mathcal{M}(\boldsymbol{x}^{(i)})(k) / \sum_{i=1}^{N} \mathbb{1}_{\{k \in \boldsymbol{y}^{(i)}\}}$
return $\mathcal{T} = \{\boldsymbol{t}_k\}$

Procedure 2: Enhance the Label Set

Input: $\mathcal{M}, \mathcal{D}, \mathcal{O}, \mathcal{T}$
Output: Enhanced Label Set $\{\boldsymbol{y}'^{(i)}\}, i \in \{1, \ldots, N\}$
Initialize $\{\boldsymbol{y}'^{(i)}\} = \{\boldsymbol{y}^{(i)}\}$
for $i \in \{1, \ldots, N\}$ **do**
  for $k \in \boldsymbol{y}^{(i)}$ **do**
    for $k_n \in \mathcal{O}(k)$ **do** ▷parent or child class of k
      if $\mathcal{M}(\boldsymbol{x}^{(i)})(k_n) > t_{k_n}$ and $k_n \notin \boldsymbol{y}^{(i)}$ **then**
        $\boldsymbol{y}'^{(i)} = \boldsymbol{y}'^{(i)} \cup \{k_n\}$
return $\{\boldsymbol{y}'^{(i)}\}$

TABLE VIII
RESULT OF LABEL ENHANCEMENT ON THE BALANCED SET (NOTE THE MAP WITHOUT LABEL ENHANCEMENT IS 0.3108±0.0013)

	Type I	Type II	Type I and II
# Impact Classes	212	93	274
Label Added (%)	3.7%	3.9%	7.2%
Impacted Class Improvement	4.5%	3.8%	4.5%
Non-impacted Class Improvement	1.9%	2.1%	1.3%
Mean Class-wise Relative Improv.	3.0%	2.4%	2.9%
mAP Improvement	1.9%	1.5%	1.7%
mAP	0.3166 ±0.0016	0.3156 ±0.0007	0.3162 ±0.0005

has an mAP of 0.3108±0.0013 (the model from the previous section). The key findings are as follows: First, a noticeable number of labels are added, and over half of the classes are impacted, which further indicates that the missing label issue is prevalent in AudioSet. Second, enhancing the label improves the performance of both impacted and non-impacted classes, but the impacted classes have a larger relative improvement. Third, the mean class-wise relative AP improvement is larger than the relative mean AP (mAP) improvement, indicating that more of the classes that improved originally had below-average performance. This supports our hypothesis that the missing label problem lowers the performance of a sound class. Fourth, we evaluate the performance of fixing Type I errors, Type II errors, and fixing both. The improvement achieved by fixing Type I errors is larger than fixing Type II errors. Fixing both cannot further improve the performance. Fifth, since the performance improvement is relatively minor, we run all experiments three times with different random seeds and report both the mean and standard deviation. As shown in the table, the results verify the statistical significance of the improvement. Finally, we also applied the label enhancement method on the full AudioSet, however, we did not observe a performance improvement. Fixing Type I, Type II, and both errors leads to mAPs of 0.4400, 0.4387, and 0.4386, respectively, while the model without label enhancement achieves an mAP of 0.4397±0.0007. We believe the main reason for the relatively small improvement achieved by label enhancement is that the same label noise exists consistently in both the training set and evaluation set. Therefore, merely applying label enhancement on the training set leads to a mismatch between the training and evaluation sets. The performance results do not therefore fully reflect the actual improvement. In addition, it is possible that the label modification threshold is not appropriate for the full AudioSet.

In order to verify these hypotheses, we evaluate our model on existing datasets with more accurate annotation including ESC-50 [1] and FSD50K [11], and also test various label modification thresholds. ESC-50 contains 2,000 audio samples of 50 sound classes, among which 40 classes are overlapped with the AudioSet. Therefore, we evaluate our model trained with AudioSet on the 1,600 samples that are labeled as these 40 overlapped classes. FSD50K is a recently collected data set of sound event audio clips with 200 classes drawn from the AudioSet ontology. The FSD50K evaluation set is more carefully annotated compared with the training and validation set and can be used as fair references. Since the length of AudioSet model input is 10s while a small portion of FSD50K audio clips are longer than 10s, we cut all FSD50K audio clips to 10s for testing. In addition, we also apply the proposed label enhancement algorithm on the AudioSet evaluation set and generate enhanced evaluation sets. We include the enhanced AudioSet evaluation sets as additional evaluation sets.

We evaluate various label modification thresholds including the mean, 25th percentile (25P), 10th percentile (10P), and 5th percentile (5P) of the teacher model's prediction score of all audio clips originally labeled as that class. The lower the threshold, the more labels are modified, e.g., using the 5th percentile of the prediction score as the threshold changes the largest number of labels. We then train models with the four enhanced label sets and compare their results on seven evaluation sets (ESC-50, FSD50K, original AudioSet evaluation set, and four enhanced AudioSet evaluation set with different label modification thresholds).

As shown in Table IX, we find that models trained with enhanced AudioSet label sets consistently outperforms the model trained with the original AudioSet label set on all evaluation sets except the original AudioSet evaluation set, demonstrating that the proposed label enhancement algorithm is able to improves the model performance, the reason why we cannot observe the improvement on the AudioSet evaluation set is that the evaluation set itself contains annotation errors. While there is no threshold that is optimal for all evaluation sets, for both balanced and full AudioSet experiments, we find the mean and

TABLE IX
AUDIOSET LABEL ENHANCEMENT (LE) EXPERIMENT RESULTS. WE USE THE MEAN, 25TH PERCENTILE (25P), 10TH PERCENTILE (10P), AND 5TH PERCENTILE (5P) OF THE PREDICTION SCORE AS THE LABEL MODIFICATION THRESHOLDS AND GENERATE 4 ENHANCED AUDIOSET TRAINING LABEL SETS AND EVALUATION LABEL SETS. WE THEN TRAIN THE MODEL WITH THE ENHANCED TRAINING SETS AND EVALUATE IT ON VARIOUS EVALUATION SETS. THE RESULTS SHOW THAT THE MODEL TRAINED WITH ENHANCED LABEL SETS CONSISTENTLY OUTPERFORMS THE MODEL TRAINED WITH ORIGINAL LABEL SETS ON ALL EVALUATION SETS EXCEPT THE ORIGINAL AUDIOSET EVALUATION SET

	Label Added (%)	ESC-50 40 Classes	FSD50k Eval	AudioSet Eval Ori	AudioSet Eval Mean	AudioSet Eval 25P	AudioSet Eval 10P	AudioSet Eval 5P
AudioSet Balanced Training Set								
No LE	0.0	0.7320	0.3443	0.3123	0.3485	0.3632	0.3540	0.3417
LE, Mean	7.2	0.7573	0.3549	0.3162	0.3591	0.3739	0.3630	0.3500
LE, 25P	22.8	**0.7639**	0.3680	**0.3165**	**0.3632**	0.3855	0.3760	0.3628
LE, 10P	44.5	0.7551	0.3639	0.3078	0.3527	0.3840	0.3811	0.3699
LE, 5P	60.2	0.7548	**0.3766**	0.3078	0.3518	**0.3862**	**0.3880**	**0.3790**
AudioSet Full Training Set								
No LE	0.0	0.8587	0.4977	**0.4397**	0.5053	0.5143	0.4930	0.4723
LE, Mean	11.1	**0.8772**	0.5079	0.4386	**0.5075**	0.5190	0.4977	0.4769
LE, 25P	37.3	0.8736	**0.5097**	0.4296	0.4999	**0.5267**	0.5093	0.4891
LE, 10P	77.7	0.8608	0.5078	0.4094	0.4752	0.5178	**0.5121**	0.4969
LE, 5P	111.9	0.8534	0.4988	0.3936	0.4560	0.5047	0.5088	**0.4987**

TABLE X
PERFORMANCE IMPACT ON MAP DUE TO WEIGHT AVERAGING

	Balanced Set	Full Set
Without Weight Averaging	0.3162 ± 0.0005	0.4397 ± 0.0007
With Weight Averaging	0.3192 ± 0.0015	0.4435 ± 0.0008

25th percentile of the teacher model's prediction score are the most appropriate label modification thresholds.

We believe it is an important and non-negligible topic for future AudioSet and general audio tagging research because noisy labels are inevitable for a large-scale dataset and errors will impact model performance. In the following section, we use models trained with the enhanced label set as default for all balanced set experiments.

VI. WEIGHT AVERAGING AND ENSEMBLE

A. Model Weight Averaging

In this section, we explore improving model performance by aggregating multiple models. The first strategy we explore is *weight averaging* [51]. Weight averaging performs an equal average of the weights traversed by the optimizer, which makes the solution fall in the center, rather than the boundary, of a wide flat low-loss region and thus lead to better generalization than conventional training. Empirically, weight averaging has been shown to improve the performance of various models such as VGG [52], ResNets [20], and DenseNets [53] on a variety of tasks [51], [54]. While weight averaging is usually applied with a high constant or cyclical learning rate, we find it is helpful even when used together with a weight decay strategy.

In this work, we simply average all weights of the model checkpoints at multiple epochs. For both balanced set and full set experiments, we start averaging model checkpoints of every epoch after the learning rate is decreased to 1/4 of the initial learning rate (i.e., the 41^{st} and the 16^{th} epochs, respectively) until the end of the training. As shown in Table X, weight averaging leads to a 0.9% improvement for both balanced set and full set experiment. We further find the improvement is not sensitive to exactly when weight averaging begins. As shown in Fig. 8, starting averaging at any epoch after the 10^{th} epochs (until the last epoch) can outperform any single checkpoint model for the full set experiment.

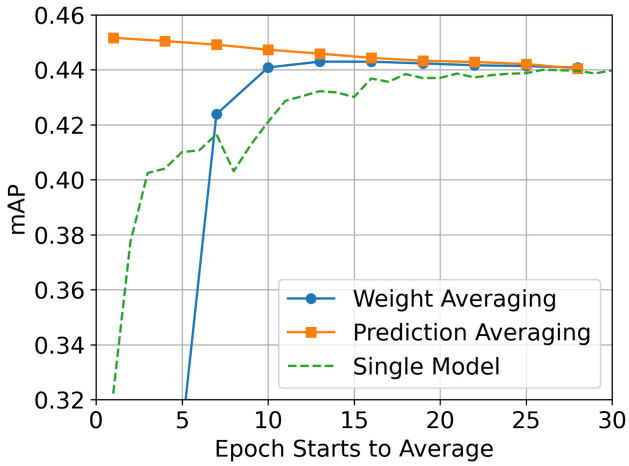

Fig. 8. Relationship of the performance of averaging models with the epoch starts to average. For both weight and prediction averaging, we average all checkpoints from the starting epoch to the last epoch, i.e., the earlier to start averaging, the more checkpoints are averaged. Note that the improvement of model averaging is not sensitive to exactly when weight averaging begins. For weight averaging, the optimal starting epoch is around the 15^{th} epoch while starting averaging at any epoch after the 10^{th} epochs can outperform any single checkpoint. For prediction averaging, starting averaging from the first epoch leads to the highest mAP, indicating averaging all checkpoints is optimal, while starting averaging at any epoch can outperform any single checkpoint. However, averaging the predictions of the last few checkpoints barely outperforms single checkpoints, indicating the importance of diversity.

In summary, weight averaging is easy to implement, adds no additional cost to training and inference, but can consistently improve model performance. By applying weight averaging to our models, we get our best single model with an mAP of

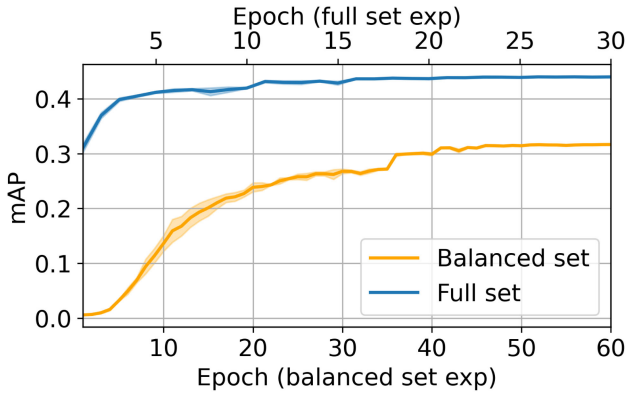

Fig. 9. The learning curve of our experiments. Each experiment is run three times, and the stand deviation is shown in the shade.

0.3192 and 0.4435 for balanced and full AudioSet experiment, respectively.

B. Ensemble

Finally, we explore a series of ensemble strategies. The goal of ensemble methods is to combine the predictions of several models to improve generalizability and robustness over any single model. Previously, ensemble of audio tagging models has been studied in in [4], [13], [15], [16], [55]–[57], but typically only one strategy is covered in each of these previous efforts. In this work, we use the simple voting algorithm, but compare multiple ways of building the model committee. The reason why we do not use iterative ensemble methods (e.g., Boosting) is because AudioSet training is expensive making iterative training computationally unreasonable for this work.

1) Checkpoint Averaging: The first strategy investigated is checkpoint averaging, whereby the output of checkpoint models at multiple epochs are averaged together. The implementation is similar to weight averaging, but is conducted in the model space rather than the weight space. Since we conduct random sampling with replacement during full set training, the combination with checkpoint averaging is the same as bootstrap aggregating (i.e., Bagging) [39]. In our experiment, we average the output of all checkpoint models (i.e., 60 and 30 checkpoint models for the balanced set and full set, respectively). As shown in the upper part of Table XI, this approach works well. Specifically, the ensembled model noticeably outperforms the best checkpoint model in the committee. In addition, as shown in Fig. 8, starting averaging from the first epoch leads to the highest mAP, indicating averaging all checkpoints is optimal. Averaging from any epoch can outperform the best single checkpoint model, which can be a simple alternative. However, this approach greatly increases the computational overhead of inference, which makes it less practical in deployment.

2) Averaging Models Trained With Different Random Seeds: Previous work suggests that ensembles generalize better when they constitute members that form a *diverse* and *accurate* set [58]. As shown in Fig. 8, starting averaging the checkpoint predictions from the last few epochs can only slightly outperform the best single checkpoint model, even though these checkpoint models are quite accurate, indicating the importance of diversity. Therefore, we run the experiment three times with the exact same setting, but with a different random seed. We then average the output of the last checkpoint model of each run. As shown in the middle part of Table XI, this approach leads to an even larger improvement than checkpoint averaging with only three models in the committee. Therefore, averaging models trained with different random seeds, while increasing the training cost (due to the repeat runs), is more practical for deployment and offers better performance.

TABLE XI
RESULTS OF MODEL ENSEMBLE. FOR EACH EXPERIMENT, WE SHOW THE NUMBER OF THE MODELS IN THE COMMITTEE (# MODELS), THE AVERAGE MAP OF MODELS IN THE COMMITTEE (AVG MAP), THE MAP OF THE BEST MODEL IN THE COMMITTEE (BEST MAP), AND THE MAP OF THE ENSEMBLE MODEL (ENSEMBLE MAP). NOTE THAT FOR ALL EXPERIMENTS, THE ENSEMBLE MAP IS HIGHER THAN THE BEST MAP

	# Models	Avg mAP	Best mAP	Ensemble mAP
Checkpoints of a Single Run				
Balanced	60	0.2369	0.3169	0.3280
Full	30	0.4236	0.4406	0.4518
Multiple Runs with Same Setting				
Balanced	3	0.3162	0.3167	0.3446
Full	3	0.4397	0.4405	0.4641
Models Trained with Different Settings				
Bal-pretrain	2	0.1978	0.2385	0.2410
Bal-mixup rate	5	0.3009	0.3123	0.3476
Bal-mixup-α	3	0.3071	0.3123	0.3418
Bal-augment	3	0.2775	0.3123	0.3281
Bal-label	4	0.3146	0.3169	0.3503
Bal-top5	5	0.3168	0.3180	0.3527
Bal-all	20	0.2987	0.3180	**0.3620**
Full-pretrain	2	0.3831	0.3939	0.4006
Full-augment	4	0.4080	0.4396	0.4578
Full-label	4	0.4397	0.4400	0.4653
Full-top5	5	0.4396	0.4405	0.4690
Full-all	10	0.4201	0.4405	**0.4744**

3) Averaging Models Trained With Different Settings: Finally, we explore averaging more models with greater diversity. Specifically, we ensemble models trained with all different settings tested in this paper, including whether pretraining is used (pretrain), different mix-up rates (mixup rate), different mix-up α (mix-up-α), different augmentation settings (augment), and different label enhancement strategies (label). As shown in the lower part of Table XI, no matter how the model committee is built, ensemble always improves the performance and outperforms the best model in the committee. In the literature, diversity is usually introduced with an intuitive motivation. For example, in [15], the authors ensemble models use different scale inputs because they believe the optimal input scale varies with the target audio events, and ensembles allows the model to extract relevant information from inputs with various scales. But according to our experimental results, the source of the diversity seems to be less important, i.e., the diversity caused by any factor is helpful for an ensemble.

In addition, we find the performance of the ensemble model is positively correlated with the accuracy of the models in the

TABLE XII
ABLATION STUDY RESULTS ON AUDIOSET

	Balanced AudioSet	Full AudioSet
PSLA Model	**0.3280**	**0.4518**
PSLA Model - Pretrain	0.2379	0.4302
PSLA Model - Balanced Sampling	-	0.3688
PSLA Model - Masking	0.3154	0.4430
PSLA Model - Mixup	0.3181	0.4493
PSLA Model - Label Enhancement	0.3229	-
PSLA Model - Ensemble	0.3162	0.4397
PSLA Model - Ensemble + WA	0.3192	0.4435

committee as well as the number of the models. For both the balanced set and full set experiments, our best model is achieved when all available models form an ensemble.

VII. SUPPLEMENTARY EXPERIMENTS

A. Ablation Study

From Section III to Section VI, we incrementally improve model performance from the baseline by incorporating a new technique with other techniques that have been found to be effective. In order to clearly identify the contribution of each technique and verify that all are necessary for the best model, we conduct an ablation study on balanced and full AudioSet. Specifically, we set the PSLA model with checkpoints ensemble as the baseline (the best model for a single training run), and then remove techniques from PSLA one by one, and check the performance. As shown in Table XII, removing any technique from PSLA leads to a performance drop, demonstrating that all proposed techniques are useful. It is worth mentioning that removing balanced sampling leads to a significant performance drop for AudioSet, the performance of the model is worse than the model only with pretraining (0.3939 mAP, in Table IV), indicating that other techniques (e.g., masking, mixup, and ensemble) should be used together with balanced sampling for AudioSet. Besides balanced sampling, removing pretraining leads to the largest performance drop, followed by ensemble, time and frequency masking, and mixup training for the full AudioSet.

B. Experiment With Various Audio Tagging Models

In the previous sections, we focus on the EfficientNet-B2 with a 4-headed attention model described in Section II-C. In order to identify if the proposed PSLA framework is model-agnostic and explore the model size-performance trade-offs, in this section, we evaluate the PSLA framework using 6 different models. All models take the same input and are trained with the same setting as mentioned in Section II-B.

1) MobileNet V2 [21]. The MobileNet model does not have an attention module. We use a fully connected layer as the classification layer.
2) EfficientNet-B0 with single-headed attention model. The model architecture is the same as the model described in Section II-C except that it is based on a smaller EfficientNet-B0 and only has one attention module.
3) EfficientNet-B2 with mean pooling model. The model architecture is the same as the model described in Section II-C except that it uses mean pooling rather than attention pooling.
4) EfficientNet-B2 with single-headed attention model. The model architecture is the same as the model described in Section II-C except that it only has one attention module.
5) EfficientNet-B2 with 4-headed attention model. This is the model we use in from Section III to Section VI and is described in Section II-C.
6) ResNet50 with single-headed attention module. This is the model proposed in [4].

To save compute, for all PSLA models, we use the checkpoint averaging ensemble that only requires a single training process, we also report the single model with weight averaging for all full AudioSet experiments. As shown in Table XIII, when trained with PSLA techniques, all models can achieve a noticeable performance improvement. This justifies that the proposed PSLA framework is model-agnostic.

Comparing the EfficientNet-B2 models with 4-headed attention, single-headed attention, and mean pooling, we find while the single 4-headed attention model performs best (0.4435 mAP), the single-headed attention model and the mean pooling model only perform slightly worse. The EfficientNet-B0 model with single-headed attention that has 5.36M parameters also achieves a comparable performance with the best existing model that has 81 M parameters [5]. The choice of the model depends on the application, e.g., attention-based models can be used for frame-level tagging; models with mean pooling can be used for streaming applications; smaller models are preferable for resource-constrained devices.

We also compute the Pearson correlation of class-wise APs between these models and find that the correlation of class-wise APs are high (over 0.95), this confirms that the poor performance of some class is not due to model architecture, but due to the data.

C. Experiment on FSD50K

In the previous sections, we focus on AudioSet. To check the generalizability of the proposed PSLA techniques, we also conduct a set of experiments on FSD50K [11]. Specifically, we train the EfficientNet-B2 model with a 4-headed attention module with an initial learning rate of 5e-4 and a batch size of 24 for 40 epochs. The learning rate is cut in half every 5 epochs after the 10^{th} epoch. Since the maximum input audio length of FSD50K is 30s, we pad all input audio clips to 30s. For the single model, we train it with the FSD50K training set, validate it on the FSD50K validation set, and evaluate it on the FSD50K evaluation set. We use the same weight averaging and checkpoint averaging ensemble setting as the AudioSet experiments. We also conduct an ablation study on FSD50K.

As shown in Table XIV, our single model, weight averaging model, and ensemble model achieve an mAP of 0.5535, 0.5571, and 0.5671 on the FSD50K evaluation set, respectively, all outperform the best existing model [59]. Removing any technique from PSLA leads to a performance drop, demonstrating that all proposed techniques can be generalized to the FSD50K dataset.

TABLE XIII
COMPARISON OF THE PERFORMANCE ON MAP OF VARIOUS MODELS TRAINED WITH PSLA AND WITHOUT PSLA ON THE BALANCED AND FULL AUDIOSET

	# Params	Balanced AudioSet			Full AudioSet		
		No PSLA	PSLA	Imp.(%)	No PSLA	PSLA	Imp.(%)
MobileNet V2	2.90M	0.1612	0.2650	64.4	0.3032	0.4058 (Single: 0.3940)	33.8
EfficientNet-B0, Single-headed Attention	5.36M	0.1529	0.3350	119.1	0.3789	0.4493 (Single: 0.4391)	18.6
EfficientNet-B2, Mean Pooling	8.44M	0.1903	0.3317	74.3	0.3325	0.4455 (Single: 0.4382)	34.0
EfficientNet-B2, Single-headed Attention	9.19M	0.1478	0.3406	130.4	0.3818	0.4556 (Single: 0.4414)	19.3
EfficientNet-B2, 4-headed Attention	13.64M	0.1570	0.3280	108.9	0.3723	0.4518 (Single: 0.4435)	21.4
ResNet-50, Single-headed Attention	25.66M	0.1635	0.3180	94.5	0.3790	0.4477 (Single: 0.4042)	18.1

TABLE XIV
EXPERIMENT RESULT ON FSD50K DATASET

	FSD50K Eval
FSD50K Baseline [11]	0.434
Audio Transformers [59]	0.537
PSLA Model	**0.5671**
PSLA Model - Pretrain	0.4524
PSLA Model - Balanced Sampling	0.5626
PSLA Model - Masking	0.5617
PSLA Model - Mixup	0.5164
PSLA Model - Label Enhancement	0.5583
PSLA Model - Ensemble	0.5535
PSLA Model - Ensemble + WA	0.5571

TABLE XV
COMPARISON WITH PREVIOUS METHODS (UPPER: BALANCED AUDIOSET EXPERIMENTS, LOWER: FULL AUDIOSET EXPERIMENTS)

	#Params	mAP	AUC	d'
Wu-minimal [60], 2018	2.6M	-	0.916	1.950
Kumar [61], 2018	-	0.213	0.927	2.056
Wu-best [60], 2018	56M	-	0.927	2.056
Kong [8], 2019	-	0.274	0.949	2.316
PANNs [5], 2020	81M	0.278	0.905	1.853
Our Baseline	13.6M	0.1570	0.9108	1.903
Proposed Single Model	13.6M	0.3192 $\pm$0.0015	0.9534 $\pm$0.0005	2.374 $\pm$0.007
Proposed 68M Model	13.6M$\times$5	0.3527	0.9602	2.479
Proposed Full Model	13.6M$\times$20	**0.3620**	**0.9638**	**2.541**
AudioSet Baseline [3]	-	0.314	0.959	2.452
Kong [6], 2018	-	0.327	0.965	2.558
Yu [7], 2018	-	0.360	0.970	2.660
TALNet [62], 2019	-	0.362	0.965	2.554
Kong [8], 2019	-	0.369	0.969	2.639
DeepRes [4], 2019	26M	0.392	0.971	2.682
PANNs [5], 2020	81M	0.439	0.973	2.725
Our Baseline	13.6M	0.3723	0.9706	2.672
Proposed Single Model	13.6M	0.4435 $\pm$0.0008	0.9753 $\pm$0.0003	2.778 $\pm$0.007
Proposed 68M Model	13.6M$\times$5	0.4690	0.9789	2.872
Proposed Full Model	13.6M$\times$10	**0.4744**	**0.9810**	**2.936**

D. Learning Curve of PSLA Models

We show the learning curve of our best single EfficientNet B2 with 4-headed attention model (without weight averaging) in Fig. 9. For both the balanced set and full set experiment, we repeat the training process three times with different random seeds and show the standard deviation in the plot. As we can see, the training converges, and the performance of the model barely varies with the random seed, i.e., the three runs achieve almost the same result.

VIII. CONCLUSION

In this paper, we describe several techniques that improve the performance of a CNN-based neural model for audio tagging. First, we show an ImageNet-pretrained CNN can noticeably improve performance. While it is straightforward to implement for CNN-based models it has seldom been used in audio tagging research. Second, due to an imbalance in sound class samples in Audioset, we describe several data balancing and augmentation strategies that alleviate the data imbalance issue and help improve performance. We argue that balanced sampling and data augmentation should be a standard component for AudioSet modeling. Third, by observing variation in class-specific performance, we identified a missing label issue with Audioset and proposed a label enhancement method that shows improvement on the balanced training set. The enhanced label set can be used in the same way as the original label set in future research. We were not able to observe a performance improvement by enhancing the full set labels, possibly due to similar missing labels in the evaluation set. Due to its impact on performance, we believe addressing the noisy label issue is an important research topic for audio tagging. Finally, we describe weight averaging and ensemble strategies that are both simple and effective for audio tagging.

By combining all these training techniques, we are able to improve the performance of a normal EfficientNet model by 130.6% and 28.2% without modifying the model architecture for the balanced and full AudioSet experiment, respectively. This magnitude of improvement is larger than was achieved by many previous model architecture or attention module development efforts, indicating that appropriate training techniques are equally important. As a consequence, by training an EfficientNet with these techniques, we obtain a single model (with 13.6 M parameters) and an ensemble model that achieve mean average precision (mAP) scores of 0.444 and 0.474 on AudioSet,

respectively, outperforming the previous best system of 0.439 with 81 M parameters [5]. Our best model trained with only the balanced AudioSet ($\sim 1\%$ of the full set) outperforms our baseline and many previous models trained with the full set. We show the AUC and d-prime of our models and compare them with previous efforts in Table XV. The proposed model outperform previous models for all evaluation metrics.

The work in this paper can serve as a recipe for AudioSet training. Most of the proposed methods are model agnostic and can be combined together with various model architectures and attention modules. As we showed in the paper, the same model can perform much better when it is trained with appropriate techniques. We hope this work can facilitate future audio tagging research by documenting a set of strong and useful training techniques.

REFERENCES

[1] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 1015–1018.

[2] P. Foster, S. Sigtia, S. Krstulovic, J. Barker, and M. D. Plumbley, "Chime-home: A dataset for sound source recognition in a domestic environment," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2015, pp. 1–5.

[3] J. F. Gemmeke *et al.*, "Audio Set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 776–780.

[4] L. Ford, H. Tang, F. Grondin, and J. R. Glass, "A deep residual network for large-scale acoustic scene analysis," in *Proc. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 2568–2572.

[5] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 2880–2894, Oct. 2020, doi: 10.1109/TASLP.2020.3030497.

[6] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, "Audio set classification with attention model: A probabilistic perspective," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 316–320.

[7] C. Yu, K. S. Barsim, Q. Kong, and B. Yang, "Multi-level attention model for weakly supervised audio classification," in *Proc. Detection Classification Acoust. Scenes Events Workshop*, 2018.

[8] Q. Kong, C. Yu, Y. Xu, T. Iqbal, W. Wang, and M. D. Plumbley, "Weakly labelled audioset tagging with attention neural networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 11, pp. 1791–1802, Nov. 2019.

[9] S. Chen, J. Chen, Q. Jin, and A. Hauptmann, "Class-aware self-attention for audio event recognition," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2018, pp. 28–36.

[10] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[11] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: An open dataset of human-labeled sound events," 2020, *arXiv:2010.00475*.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[13] K. Palanisamy, D. Singhania, and A. Yao, "Rethinking CNN models for audio classification," 2020, *arXiv:2007.11154*.

[14] E. Fonseca, S. Hershey, M. Plakal, D. P. Ellis, A. Jansen, and R. C. Moore, "Addressing missing labels in large-scale sound event recognition using a teacher-student framework with loss masking," *IEEE Signal Process. Lett.*, vol. 27, pp. 1235–1239, Jul. 2020, doi: 10.1109/LSP.2020.3006378.

[15] D. Lee, S. Lee, Y. Han, and K. Lee, "Ensemble of convolutional neural networks for weakly-supervised sound event detection using multiple scale input," in *Proc. Detection Classification Acoust. Scenes Events Workshop*, 2017.

[16] P. Lopez-Meyer, J. A. del H. Ontiveros, G. Stemmer, L. Nachman, and J. Huang, "Ensemble of convolutional neural networks for the DCASE 2020 acoustic scene classification challenge," in *Proc. Detection Classification Acoust. Scenes Events Workshop*, 2020.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.

[18] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, 1997.

[19] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 233–240.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.

[22] M. Tan *et al.*, "Mnasnet: Platform-aware neural architecture search for mobile," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2820–2828.

[23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. North Amer. Chapter Assoc. Computat. Linguistics-Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[24] K. He, R. Girshick, and P. Dollár, "Rethinking ImageNet pre-training," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4918–4927.

[25] Y.-A. Chung, W.-N. Hsu, H. Tang, and J. Glass, "An unsupervised autoregressive model for speech representation learning," in *Proc. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 146–150.

[26] A. Saeed, D. Grangier, and N. Zeghidour, "Contrastive learning of general-purpose audio representations," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 3875–3879.

[27] J. Shor *et al.*, "Towards learning a universal non-semantic representation of speech," in *Proc. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 140–144.

[28] M. Tagliasacchi, B. Gfeller, F. D. C. Quitry, and D. Roblek, "Pre-training audio representations with self-supervision," *IEEE Signal Process. Lett.*, vol. 27, pp. 600–604, Apr. 2020, doi: 10.1109/LSP.2020.2985586.

[29] A. Jansen *et al.*, "Unsupervised learning of semantic audio representations," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 126–130.

[30] L. Wang, K. Kawakami, and A. van den Oord, "Contrastive predictive coding of audio with an adversary," in *Proc. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 826–830.

[31] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 1041–1044.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.

[33] G. Gwardys and D. M. Grzywczak, "Deep image features in music information retrieval," *Int. J. Electron. Telecommun.*, vol. 60, no. 4, pp. 321–326, 2014.

[34] A. Guzhov, F. Raue, J. Hees, and A. Dengel, "ESResNet: Environmental sound classification based on visual domain models," in *Proc. 25th Int. Conf. Pattern Recognit.*, 2021, pp. 4933–4940.

[35] S. Adapa, "Urban sound tagging using convolutional neural networks," in *Proc. Detection Classification Acoust. Scenes Events Workshop*, 2019, pp. 5–9.

[36] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[37] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[38] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. Boca Raton, FL, USA: CRC Press, 1994.

[39] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[40] D. S. Park *et al.*, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 2613–2617.

[41] Y. Tokozume, Y. Ushiku, and T. Harada, "Between-class learning for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5486–5494.

[42] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, " Mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Representations*, 2018.

[43] Y. Tokozume, Y. Ushiku, and T. Harada, "Learning from between-class examples for deep sound recognition," in *Proc. Int. Conf. Learn. Representations*, 2018.

[44] A. Mesaros *et al.*, "DCASE 2017 challenge setup: Tasks, datasets and baseline system," in *Proc. Detection Classification Acoust. Scenes Events Workshop*, 2017, pp. 85–92.
[45] A. Shah, A. Kumar, A. G. Hauptmann, and B. Raj, "A closer look at weak label learning for audio events," 2018, *arXiv:1804.09288*.
[46] L. H. Ford, "Large-scale acoustic scene analysis with deep residual networks," Master's thesis, Massachusetts Inst. Technol., 2019.
[47] K. Wu and D. G. Childers, "Gender recognition from speech. Part I: Coarse analysis," *J. Acoust. Soc. Amer.*, vol. 90, no. 4, pp. 1828–1840, 1991.
[48] D. G. Childers and K. Wu, "Gender recognition from speech. Part II: Fine analysis," *J. Acoust. Soc. Amer.*, vol. 90, no. 4, pp. 1841–1856, 1991.
[49] Z.-Q. Wang and I. Tashev, "Learning utterance-level representations for speech emotion and age/gender recognition using deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 5150–5154.
[50] M. Meire, L. Vuegen, and P. Karsmakers, "The impact of missing labels and overlapping sound events on multi-label multi-instance learning for sound event classification," in *Proc. Detection Classification Acoust. Scenes Events Workshop*, 2019, pp. 159–163.
[51] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," in *Proc. 34th Conf. Uncertainty Artif. Intell.*, 2018, pp. 876–885.
[52] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition,"in *Proc. Int. Conf. Learn. Representations*, 2015.
[53] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.
[54] B. Athiwaratkun, M. Finzi, P. Izmailov, and A. G. Wilson, "There are many consistent explanations of unlabeled data: Why you should average," in *Proc. Int. Conf. Learn. Representations*, 2019.
[55] Z. Shi, L. Liu, H. Lin, R. Liu, A. Shi, and S. First, "HODGEPODGE: Sound event detection based on ensemble of semi-supervised learning methods," in *Proc. Detection Classification Acoust. Scenes Events Workshop*, 2019, pp. 224–228.
[56] Y. Guo, M. Xu, Z. Wu, J. Wu, and B. Su, "Multi-scale convolutional recurrent neural network with ensemble method for weakly labeled sound event detection," in *Proc. 8th Int. Conf. Affect. Comput. Intell. Interact. Workshops Demos*, 2019, pp. 1–5.
[57] W. Lim, S. Suh, S. Park, and Y. Jeong, "Sound event detection in domestic environments using ensemble of convolutional recurrent neural networks," in *Proc. Detection Classification Acoust. Scenes Events Workshop*, 2019.
[58] A. Chandra, H. Chen, and X. Yao, "Trade-off between diversity and accuracy in ensemble generation," in *Multi-Objective Machine Learning*. Berlin, Germany: Springer, 2006, pp. 429–464.
[59] P. Verma and J. Berger, "Audio transformers: Transformer architectures for large scale audio understanding. Adieu convolutions," 2021, *arXiv:2105.00335*.
[60] Y. Wu and T. Lee, "Reducing model complexity for DNN based large-scale audio classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 331–335.
[61] A. Kumar, M. Khadkevich, and C. Fügen, "Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 326–330.
[62] Y. Wang, J. Li, and F. Metze, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 31–35.

Yuan Gong (Member, IEEE) received the B.S. degree in biomedical engineering from Fudan University, Shanghai, China, and the Ph.D. degree in computer science from the University of Notre Dame, IN, USA, in 2015 and 2020, respectively. He is a Postdoctoral Researcher with the MIT Computer Science and Artificial Intelligence Laboratory. Currently, his primary research interests include automatic speech recognition, speech based healthcare system, and acoustic events detection. He won the 2017 AVEC depression detection challenge and one of his papers was nominated for the Best Student Paper Award in Interspeech 2019.

Yu-An Chung (Member, IEEE) received the B.S. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, and the S.M. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2016 and 2019, respectively. He is a Ph.D. Candidate working with Dr. James Glass in the MIT Computer Science and Artificial Intelligence Laboratory. His current research interests include methods for self-supervised learning of speech and language representations. His primary research goal is to develop spoken language technology using as few human-annotated data as possible.

James Glass (Fellow, IEEE) is a Senior Research Scientist with MIT where he Leads the Spoken Language Systems Group with the Computer Science and Artificial Intelligence Laboratory. He is also a member of the Harvard University Program in Speech and Hearing Bioscience and Technology. Since obtaining the S.M. and Ph.D. degrees with MIT in electrical engineering and computer science, his research interests include automatic speech recognition, unsupervised speech processing, and spoken language understanding. He is a Fellow of the International Speech Communication Association, and is currently an Associate Editor for the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.