# Weakly Labelled AudioSet Tagging with Attention Neural Networks

Qiuqiang Kong, *Student Member, IEEE,*, Changsong Yu, Yong Xu, Turab Iqbal, Wenwu Wang, Mark D. Plumbley

***Abstract*—Audio tagging is the task of predicting the presence or absence of sound classes within an audio clip. Previous work in audio tagging focused on relatively small datasets limited to recognising a small number of sound classes. We investigate audio tagging on AudioSet, which is a dataset consisting of over 2 million audio clips and 527 classes. AudioSet is weakly labelled, in that only the presence or absence of sound classes is known for each clip, while the onset and offset times are unknown. To address the weakly-labelled audio tagging problem, we propose attention neural networks as a way to attend the most salient parts of an audio clip. We bridge the connection between attention neural networks and multiple instance learning (MIL) methods, and propose decision-level and feature-level attention neural networks for audio tagging. We investigate attention neural networks modelled by different functions, depths and widths. Experiments on AudioSet show that the feature-level attention neural network achieves a state-of-the-art mean average precision (mAP) of 0.369, outperforming the best multiple instance learning (MIL) method of 0.317 and Google's deep neural network baseline of 0.314. In addition, we discover that the audio tagging performance on AudioSet embedding features has a weak correlation with the number of training examples and the quality of labels of each sound class.**

***Index Terms*—Audio tagging, AudioSet, attention neural network, weakly labelled data, multiple instance learning.**

## I. Introduction

Audio tagging is the task of predicting the tags of an audio clip. R2: Audio tagging is a multi-class tagging problem to predict zero, one or multiple tags for an audio clip. As a specific task of audio tagging, audio scene classification often involves the prediction of only one label in an audio clip, i.e. the type of environment in which the sound is present. In this paper, we focus on audio tagging. Audio tagging has many applications such as music tagging [1] and information retrieval [2]. An example of audio tagging that has attracted significant attention in recent years is the classification of environmental sounds, that is, predicting the scenes where they are recorded. For instance, the Detection and Classification of Acoustic Scenes and Events (DCASE) challenges [3]–[6] consist of tasks from a variety of domains, such as DCASE 2018 Task 1 classification of outdoor sounds, DCASE 2017 Task4 tagging of street sounds and DCASE 2016 Task4 tagging of domestic sounds. These challenges provide labelled datasets, so it is possible to use supervised learning algorithms for audio tagging. However, many audio tagging datasets are relatively small [3]–[6], ranging from hundreds to thousands of training examples, while modern machine learning methods such as deep learning [7, 8] often benefit greatly from larger dataset for training.

In 2017, a large-scale dataset called *AudioSet* [9] was released by Google. AudioSet consists of audio clips extracted from YouTube videos, and is the first dataset that achieves a similar scale to the well-known ImageNet [10] dataset in computer vision. The current version (v1) of AudioSet consists of 2,084,320 audio clips organised into a hierarchical ontology with 527 predefined sound classes in total. Each audio clip in AudioSet is approximately 10 seconds in length, leading to 5800 hours of audio in total. AudioSet provides an opportunity for researchers to investigate a large and broad variety of sounds instead of being limited to small datasets with limited data and sound classes.

One challenge of AudioSet tagging is that AudioSet is a weakly-labelled dataset (WLD) [11, 12]. That is, for each audio clip in the dataset, only the presence or the absence of sound classes is indicated, while the onset and offset times are unknown. R2: In previous work in audio tagging, an audio clip is usually split into segments and each segment is assigned with the label of the audio clip [13]. However, as the onset and offset of sound events are unknown so such label assignment can be incorrect. For example, a transient sound event may only appear a short time in a long audio recording. The duration of sound events can be very different and there is no prior knowledge of their duration. Different from ImageNet [10] for image classification where objects are usually centered and have similar scale, in AudioSet the duration of sound events may vary a lot. To illustrate, Fig. 1 from top to bottom shows: the log mel spectrogram of a 10-second audio clip[1]; AudioSet bottleneck features [9] extracted by a pre-trained VGGish convolutional network followed by a principal component analysis (PCA); weak labels of the audio including "music", "chuckle", "snicker" and "speech'. In contrast to WLD, strongly labelled data (SLD) refers to the data labelled with both the presence of sound classes as well as their onset and offset times. For example, the sound event detection tasks in DCASE challenge 2013, 2016, 2017 [3, 5, 6] provide SLD. However, labelling onset

This research was supported by EPSRC grant EP/N014111/1 "Making Sense of Sounds" and a Research Scholarship from the China Scholarship Council (CSC) No. 201406150082. *(First author: Qiuqiang Kong.) (Corresponding author: Yong Xu).*

Qiuqiang Kong, Turab Iqbal, Wenwu Wang and Mark D. Plumbley are with the Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, GU2 7XH, UK. Wenwu Wang is also with Qingdao University of Science and Technology, China. (e-mail: q.kong@surrey.ac.uk; t.iqbal@surrey.ac.uk; w.wang@surrey.ac.uk; m.plumbley@surrey.ac.uk)

Changsong Yu is with University of Stuttgart, Germany. (e-mail: changsong_yu@yahoo.de)

Yong Xu is with the Tencent AI Lab, Bellevue, USA. (e-mail: lucayongxu@tencent.com)

[1]https://www.youtube.com/embed/Wxa36SSZx8o?start=70&end=80

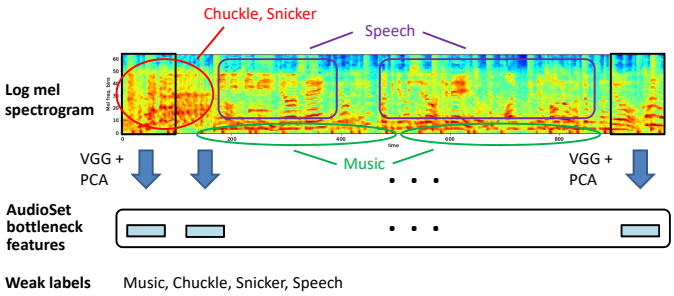


Fig. 1. From top to bottom: Log mel spectrogram of a 10-second audio clip; AudioSet bottleneck features extracted by a pre-trained VGGish convolutional neural network followed by a principle component analysis (PCA) [14]; Weak labels of the audio clip. There are no onset and offset times of the sound classes.

and offset times of sound events is time-consuming, so these strongly labelled datasets are usually limited to a relative small sample [3, 5, 6], which may limit the performance of deep neural networks that require large data to train a good model.

In this paper, we train an audio tagging system on the large-scale weakly labelled AudioSet. We bridge our previously proposed attention neural networks [15, 16] with multiple instance learning (MIL) [17] and propose decision-level and feature-level attention neural networks for audio tagging. The contributions of this paper include the following:

- Decision-level and feature-level attention neural networks are proposed for audio tagging;
- Attention neural networks modelled by different functions, widths and depth are investigated;
- The impact of the number of training examples per class on the audio tagging performance is studied;
- The impact of the quality of labels on the audio tagging performance is studied.

This paper is organised as follows. Section II introduces audio tagging with weakly labelled data. Section III introduces our previously proposed attention neural networks [15, 16]. Section IV introduces multiple instance learning. Section V reviews attention neural networks under the MIL framework and proposes decision-level and feature-level attention models. Section VI shows the experimental results. Section VII concludes and forecasts future work.

## II. Audio Tagging with Weakly Labelled Data

Audio tagging has attracted much research interests in recent years. For example, the tagging of the CHiME Home dataset [18], the UrbanSound dataset [19] and datasets from the Detection and Classification of Acoustic Scenes and Events (DCASE) challenges in 2013 [20], 2016 [21], 2017 [6] and 2018 [22]. The DCASE 2018 Challenge includes acoustic scene classification [22], general purpose audio tagging [23] and bird audio detection [24] tasks. Mel frequency cepstral coefficients (MFCC) [25]–[27] have been widely used as features to build audio tagging systems. Other features used for audio tagging include pitch features [26] and I-vectors [27]. Gaussian mixture models (GMMs) [28] and support vector machines [29]. Recently, neural networks have been used for audio tagging with mel spectrograms as input features. A variety of neural network methods including fully-connected neural networks [13], convolutional neural networks (CNNs) [14, 30, 31] and convolutional recurrent neural networks (CRNNs) [32, 33] have been explored for audio tagging. R1: An identify, locate and separate model [34] was proposed for audio-visual object extraction in large video collections using weak supervision.

A WLD consists of a set of *bags*, where each bag is a collection of instances. For a particular sound class, a positive bag contains at least one positive instance, while a negative bag contains no positive instances. We denote the $n$-th bag in the dataset as $B_n = \{\mathbf{x}_{n1}, ..., \mathbf{x}_{nT_n}\}$, where $T_n$ is the number of instances in the bag. An instance $\mathbf{x}_{nt} \in \mathbb{R}^M$ in the bag has a dimension of $M$. A WLD can be denoted as $D = \{B_n, \mathbf{y}_n\}_{n=1}^N$, where $\mathbf{y}_n \in \{0,1\}^K$ denotes the tags of bag $B_n$ and $K$ and $N$ are the number of classes and training examples, respectively. In WLD, each bag $B_n$ has associated tags but we do not know the tags of individual instances $\mathbf{x}_{nt}$ within the bag [35]. For example, in the AudioSet dataset, a bag consists of instances that are bottleneck features obtained by inputting a logmel to a pre-trained VGGish convolutional neural network. In the following sections, we omit the training example index $n$ and the time index $t$ to simplify notation.

Previous audio tagging systems using WLD have been based on segment based methods. Each segment is called an instance and are assigned the tags inherited from the audio clip. During training, instance-level classifiers are trained on individual instances. During inference, bag-level predictions are obtained by aggregating the instance-level predictions [13]. Recently, convolutional neural networks have been applied to audio tagging [32], where the log spectrogram of an audio clip is used as input to a CNN classifier without predicting the individual instances explicitly. Attention neural networks have been proposed for AudioSet tagging in [15, 16]. Later, a clip-level and segment-level model with attention supervision was proposed in [36].

## III. Audio Tagging with Attention Neural Networks

### A. Segment based methods

R3: In segment based methods, an audio clip is split into segments and each segment is assigned the tags inherited from the audio clip. In MIL, each segment is called an instance. An instance-level classifier $f$ is trained on the individual instances: $f : \mathbf{x} \mapsto f(\mathbf{x})$, where $f(\mathbf{x}) \in [0,1]^K$ predicts the presence probabilities of sound classes. The function $f$ depends on a set of learnable parameters that can be optimised using gradient descent methods with the loss function

$$l(f(\mathbf{x}), \mathbf{y}) = d(f(\mathbf{x}), \mathbf{y}), \tag{1}$$

where $\mathbf{y} \in \{0,1\}^K$ are the tags of the instance $\mathbf{x}$ and $d(\cdot,\cdot)$ is a loss function. For instance, it could be binary cross-entropy for multi-class tagging, given by

$$d(f(\mathbf{x}), \mathbf{y}) = -\sum_{k=1}^{K}[y_k \log f(\mathbf{x})_k + (1-y_k)\log(1-f(\mathbf{x})_k]. \tag{2}$$

In inference, the prediction of a bag is obtained by aggregating the predictions of individual instances in the bag such as by majority voting [13]. The segment based model has been

applied to many tasks such as information retrieval [37] due to its simplicity and efficiency. However, the assumption that all instances inherit the tags of a bag is incorrect. For example, some sound events may only occur for a short time in an audio clip.

### B. Attention neural networks

Attention neural networks were first proposed for natural language processing [38, 39], where the words in a sentence are attended differently for machine translation. Attention neural networks are designed to attend to important words and ignore irrelevant words. Attention models have also been applied to computer vision, such as image captioning [40] and information retrieval [41]. We proposed attention neural networks for audio tagging and sound event detection with WLD in [15, 33]: these were ranked first in the DCASE 2017 Task 4 challenge [33]. In a similar way to the segment based model, attention neural networks build an instance-level classifier $f(\mathbf{x})$ for individual instances $\mathbf{x}$. In contrast to the segment based model, attention neural networks do not assume that instances in a bag have the same tags as the bag. As a result, there is no instance-level ground truth for supervised learning using (1). To solve this problem, we aggregate the instance-level predictions $f(\mathbf{x})$ to a bag-level prediction $F(B)$ given by

$$F(B)_k = \sum_{\mathbf{x}\in B} p(\mathbf{x})_k f(\mathbf{x})_k, \tag{3}$$

where $p(\mathbf{x})_k$ is a weight of $f(\mathbf{x})_k$ that we refer to as an *attention function*. The attention function $p(\mathbf{x})_k$ should satisfy

$$\sum_{\mathbf{x}\in B} p(\mathbf{x})_k = 1, \tag{4}$$

so that the bag-level prediction can be seen as a weighted sum of the instance-level predictions. Both the attention function $p(\mathbf{x})$ and the instance-level classifier $f(\mathbf{x})$ depend on a set of learnable parameters. The attention function $p(\mathbf{x})_k$ controls how much a prediction $f(\mathbf{x})_k$ should be attended. Large $p(\mathbf{x})_k$ indicates that $f(\mathbf{x})_k$ should be attended, while small $p(\mathbf{x})_k$ indicates that $f(\mathbf{x})_k$ should be ignored. To satisfy (4), the attention function $p(\mathbf{x})_k$ can be modelled as

$$p(\mathbf{x})_k = v(\mathbf{x})_k / \sum_{\mathbf{x}\in B} v(\mathbf{x})_k, \tag{5}$$

where $v(\cdot)$ can be any non-negative function to ensure that $p(\cdot)$ is a probability.

An extension of the attention neural network in (3) is the multi-level attention model [16], where multiple attention modules are applied to utilise the hierarchical information of neural networks:

$$F(B) = g(F_1(B), ..., F_L(B)), \tag{6}$$

where $F_l(B)$ is the output of the $l$-th attention module and $L$ is the number of attention modules. Then a mapping $g$ is used to map from the predictions of $L$ attention modules to the final prediction of a bag. The multi-level attention neural network has achieved state-of-the-art performance in AudioSet tagging.

In the next section, we show that the attention neural networks explored above can be categorised into an MIL framework.

## IV. Multiple instance learning

Multiple instance learning (MIL) [17, 42] is a type of supervised learning method. Instead of receiving a set of labelled instances, the learner receives a set of labelled bags. MIL methods have many applications. For example, in [42], MIL is used to predict whether new molecules are qualified to make some new drug, where molecules may have many alternative low-energy states, but only one, or some of them, are qualified to make a drug. Inspired by the MIL methods, a sound event detection system trained on WLD [11] was proposed. General MIL methods include the expectation-maximization diversity density (EM-DD) method [43], support vector machine (SVM) methods [44] and neural network MIL methods [45, 46]. In [47], several MIL pooling methods were investigated in audio tagging. Attention-based deep multiple instance learning is proposed in [48].

In [35], MIL methods are grouped into three categories: the instance space (IS) methods, where the discriminative information is considered to lie at the instance-level; the bag space (BS) methods, where the discriminative information is considered to lie at the bag-level; and the embedded space (ES) methods, where each bag is mapped to a single feature vector that summarises the relevant information about a bag. We introduce the IS, BS and ES methods in more detail below.

### A. Instance space methods

In IS methods, an instance-level classifier $f : \mathbf{x} \mapsto f(\mathbf{x})$ is used to predict the tags of an instance $\mathbf{x}$, where $f(\mathbf{x}) \in [0, 1]^K$ predicts the presence probabilities of sound classes. The IS methods introduce aggregation functions [35] to convert an instance-level classifier $f$ to a bag-level classifier $F : B \mapsto [0, 1]^K$, given by

$$F(B) = \text{agg}\left(\{f(\mathbf{x})\}_{\mathbf{x}\in B}\right), \tag{7}$$

where $\text{agg}(\cdot)$ is an aggregation function. The classifier $f$ depends on a set of learnable parameters. When the IS method is trained with (1) in which each instance inherits the tags of the bag, the IS method is equivalent to the segment based model. On the other hand, the IS method can also be trained using the bag-level loss function:

$$l(F(B), \mathbf{y}) = d(F(B), \mathbf{y}), \tag{8}$$

where $\mathbf{y} \in \{0, 1\}^K$ is the tag of the bag and $d(\cdot, \cdot)$ is a loss function such as the binary cross-entropy in (2).

To model the aggregation function, the standard multiple instance (SMI) assumption and collective assumption (CA) are proposed in [35]. Under the SMI assumption, a bag-level classifier can be obtained by

$$F(B)_k = \max_{\mathbf{x}\in B} f(\mathbf{x})_k, \tag{9}$$

where the subscript $k$ denotes the $k$-th sound class of the instance-level prediction $f(\mathbf{x})$ and the bag-level prediction $F(B)$. Under the SMI assumption, for the $k$-th sound class, only one instance with the maximum prediction probability is chosen as a positive instance.

One problem of the SMI assumption is that a positive bag may contain more than one positive instance. In SED, some

sound classes such as "ambulance siren" may last for several seconds and may occur in many instances. In contrast to the SMI assumption, with the CA assumption, all the instances in a bag contribute equally to the tags of the bag. The bag-level prediction can be obtained by averaging the instance-level predictions:

$$F(B) = \frac{1}{|B|} \sum_{\mathbf{x} \in B} f(\mathbf{x}). \tag{10}$$

The symbol $|B|$ denotes the number of instances in bag $B$. Equation (10) shows that CA is based on the assumption that all the instances in a positive bag are positive.

### B. Bag space methods

Instead of building an instance-level classifier, the BS methods regard a bag $B$ as an entirety. Building a tagging model on the bags rely on a distance function $D(\cdot, \cdot) : B \times B \mapsto \mathbb{R}$. The distance function can be, for example, the Hausdorff distance [49]:

$$D(B_1, B_2) = \min_{\mathbf{x}_1 \in B_1, \mathbf{x}_2 \in B_2} \|\mathbf{x}_1 - \mathbf{x}_2\|. \tag{11}$$

In (11), the distance between two bags is the minimum distance between the instances in bag $B_1$ and $B_2$. Then this distance function can be plugged into a standard distance-based classifier such as a k-nearest neighbour (KNN) or a support vector machine (SVM) algorithm. The computational complexity of (11) is $|B_1||B_2|$, which is larger than the IS and the ES methods described below.

### C. Embedded space methods

Different from the IS methods, ES methods do not classify individual instances. Instead, the ES methods define an embedding mapping from a bag to an embedding vector:

$$f_{\text{emb}} : B \mapsto \mathbf{h}. \tag{12}$$

Then the tags of a bag is obtained by applying a function $g$ on the embedding vector:

$$F(B) = g(\mathbf{h}). \tag{13}$$

The embedding mapping $f_{\text{emb}}$ can be modelled in many ways. For example, by averaging the instances in a bag, as in the simple MI method in [50]:

$$\mathbf{h} = \frac{1}{|B|} \sum_{\mathbf{x} \in B} \mathbf{x}. \tag{14}$$

Alternatively, the mapping can be obtained in terms of the max-min operations on the instances [51]:

$$\begin{cases} \mathbf{h} = (a_1, ..., a_M, b_1, ..., b_M), \\ a_m = \max\limits_{\mathbf{x} \in B}(x_m), \\ b_m = \min\limits_{\mathbf{x} \in B}(x_m), \end{cases} \tag{15}$$

where $x_m$ is the $m$-th dimension of $\mathbf{x}$. Equation (15) shows that only one instance with the maximum or the minimum value is chosen for each dimension, while other instances have no contribution to the embedding vector $\mathbf{h}$. The ES methods summarise a bag containing an arbitrary number of instances with a vector of fixed size. Similar methods have been proposed in natural language processing to summarise sentences with a variable number of words [52].

## V. Attention neural networks under MIL

In this section, we show that the previously proposed attention neural networks [15, 16] belong to MIL frameworks, especially the IS methods. We refer to these attention neural networks as decision-level attention neural networks, because the prediction of a bag is obtained by aggregating the predictions of instances (see (7)). We then propose feature-level attention neural networks inspired by the ES methods with attention in the hidden layers.

### A. Decision-level attention neural networks

The IS methods predict the tags of a bag by aggregating the predictions of individual instances in the bag described in (7). Section IV-A shows that conventional IS methods are based on either the SMI assumption (see (9)) or the CA (see (10)). The problem of the SMI assumption is that only one instance in a bag is considered to be positive for a sound class while other instances are not considered. The SMI assumption is not appropriate for bags with more than one positive instance for a sound class. On the other hand, CA assumes that all instances in a positive bag are positive. CA is not appropriate for sound events that only last for a short time. To address the problems of the SMI and CA methods, a decision-level attention neural network based on the IS methods in (7) is proposed to learn an attention function to weight the predictions of instances in a bag, so that

$$\begin{aligned} F(B)_k &= \text{agg}(\{f(\mathbf{x})_k\}_{\mathbf{x} \in B}) \\ &= \sum_{\mathbf{x} \in B} p(\mathbf{x})_k f(\mathbf{x})_k, \end{aligned} \tag{16}$$

where $p(\mathbf{x})$ is an attention function modelled by (5). We refer to (16) as a decision-level attention neural network because the attention function $p(\mathbf{x})$ is multiplied with the predictions of the instances $f(\mathbf{x})$ to obtain the bag-level prediction. The attention function $p(\mathbf{x})$ controls how much the prediction of an instance $f(\mathbf{x})$ should be attended or ignored. Equation (16) can be seen as a general case of the SMI and CA assumptions. When one instance $\mathbf{x}$ in a bag has a value of $p(\mathbf{x}) = 1$ and another has a value of $p(\mathbf{x}) = 0$, then (16) is equivalent to the SMI assumption in (9). When $p(\mathbf{x}) = \frac{1}{|B|}$ for all instances in a bag, (16) is equivalent to CA.

Fig. 2 shows different ways to model the attention neural network in (16). For example, Fig. 2(a) shows the joint detection and classification (JDC) model [12] with attention function $p$ and the classifier $f$ modelled by separate branches. Fig. 2(b) shows the self attention neural network [53] proposed in natural language processing. Fig. 2 shows the JDC improved by using shared layers for the attention function $p$ and the classifier $f$ before they bifurcate in the penultimate layer [15].

In the attention neural networks, both $p$ and $f$ depend on a set of learnable parameters which can be optimised with gradient descent methods using the loss function in (8). For the

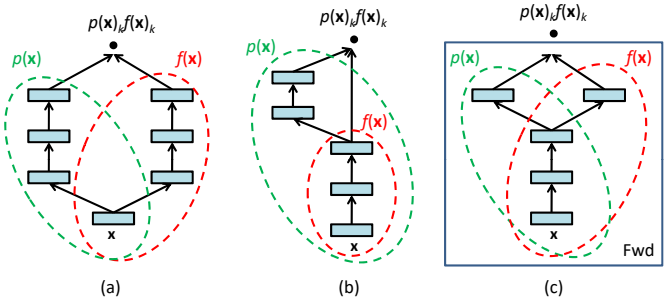


Fig. 2. (a) Joint detection and classification (JDC) model; (b) Self attention neural network in [53]; (c) Proposed attention neural network [15].

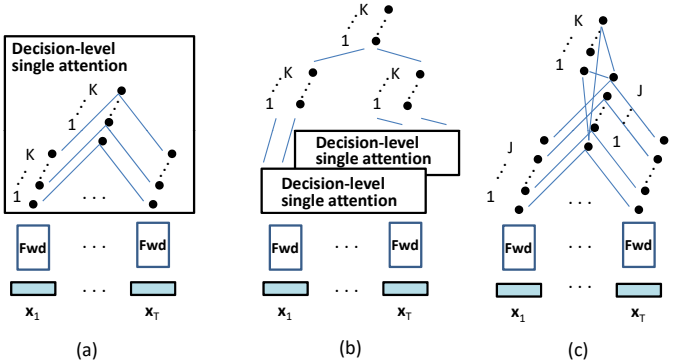


Fig. 3. (a) Decision-level single attention neural network [15]; (b) Decision-level multiple attention neural network [16]; (c) Feature-level attention neural network (proposed).

proposed model in Fig. 2(c), the attention function $p$ and the classifier $f$ share the low-level layers. We denote the output of the layer before they bifurcate as $\mathbf{x}'$. The mapping from $\mathbf{x}$ to $\mathbf{x}'$ can be modelled by fully-connected layers, for example.

$$\mathbf{x}' = f_{\text{FC}}(\mathbf{x}). \tag{17}$$

The classifier $f$ can be modelled by

$$f(\mathbf{x}) = \sigma(\mathbf{W}_1\mathbf{x}' + \mathbf{b}_1), \tag{18}$$

where $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function. The attention function $p$ can be modelled by

$$\begin{cases} v(\mathbf{x}')_k = \phi_1(\mathbf{U}_1\mathbf{x}' + \mathbf{c}_1), \\ p(\mathbf{x})_k = v(\mathbf{x}')_k / \sum_{\mathbf{x}\in B} v(\mathbf{x}')_k, \end{cases} \tag{19}$$

where $\phi_1$ can be any non-negative function to ensure $p(\mathbf{x})_k$ is a probability.

### B. Feature-level attention neural network

The limitation of the decision-level attention neural networks is that the attention function $p(\mathbf{x})$ is only applied to the prediction of the instances $f(\mathbf{x})$, as shown in (16). In this section, we propose to apply attention to the hidden layers of a neural network. This is inspired by the ES methods in (12), where a bag $B$ is mapped to a fixed-size vector $\mathbf{h}$ before being classified. We model (12) with attention aggregation:

$$h_j = \sum_{\mathbf{x}\in B} q(\mathbf{x})_j u(\mathbf{x})_j, \tag{20}$$

where both $q(\mathbf{x}) \in [0,1]^J$ and $u(\mathbf{x}) \in \mathbb{R}^J$ have a dimension of $J$. The embedded vector $\mathbf{h} \in \mathbb{R}^J$ summarises the information of a bag. Then the tags of a bag $B$ can be obtained by classifying the embedding vector:

$$F(B) = f(\mathbf{h}). \tag{21}$$

The probability $q(\mathbf{x})_j$ in (20) is the attention function of $u(\mathbf{x})_j$ and should satisfy

$$\sum_{\mathbf{x}\in B} q(\mathbf{x})_j = 1. \tag{22}$$

We model $u(\mathbf{x})$ with

$$u(\mathbf{x}) = \psi(\mathbf{W}_2\mathbf{x}' + \mathbf{b}_1), \tag{23}$$

where $\psi$ can be any linear or non-linear function to increase the representation ability of the model. The attention function $q$ can be modelled by

$$\begin{cases} w(\mathbf{x}')_j = \phi_2(\mathbf{U}_2\mathbf{x}' + \mathbf{c}_2), \\ q(\mathbf{x})_j = w(\mathbf{x}')_j / \sum_{\mathbf{x}\in B} w(\mathbf{x}')_j, \end{cases} \tag{24}$$

where $w(\mathbf{x})_j$ can be any non-negative function to ensure $q(\mathbf{x})_j$ is a probability.

Fig. 3 shows the decision-level single attention [15], decision-level multiple attention [16] and the proposed feature-level attention neural network. The forward (Fwd) block in Fig. 3 is the same as the block in Fig. 2(c). The difference between the feature-level attention function $q(\mathbf{x})$ and the decision-level attention function $p(\mathbf{x})$ is that the dimension of $q(\mathbf{x})$ can be any value, while the dimension of $p(\mathbf{x})$ is fixed to be the number of sound classes $K$. Therefore, the capacity of the decision-level attention neural networks is limited. With an increase in the dimension of $q(\mathbf{x})$, the capacity of feature-level attention neural networks is increased. The decision-level attention function attends to the predictions of instances, while the feature-level attention function attends to the features, so it is equivalent to feature selection. The multi-level attention model [16] in (6) can be seen as a special case of the feature-level attention model, with embedding vector $\mathbf{h} = (F_1(B), ..., F_L(B))$. The superior performance of the multi-level attention model shows that the feature-level attention neural networks have the potential to perform better than the decision-level attention neural networks.

### C. Modeling the attention function with different non-linearity

We adopt Fig. 2(c) as the backbone of our attention neural networks. The attention function $p$ and $q$ for the decision-level and feature-level attention neural networks are obtained via non-negative functions $\phi_1$ and $\phi_2$, respectively. The $\phi_1$ and $\phi_2$ appearing in the summation term of the denominator of (19) and (24) may affect the optimisation of the attention neural networks. We investigate modelling $\phi_2$ in the feature-level attention neural networks with different non-negative functions, including ReLU [54], exponential, sigmoid, softmax and network-in-network (NIN) [55]. We omit the evaluation of $\phi_1$, as the feature-level attention neural networks outperform the decision-level attention neural networks. The ReLU function is defined as [54]

$$\phi(z) = \max(z, 0). \tag{25}$$

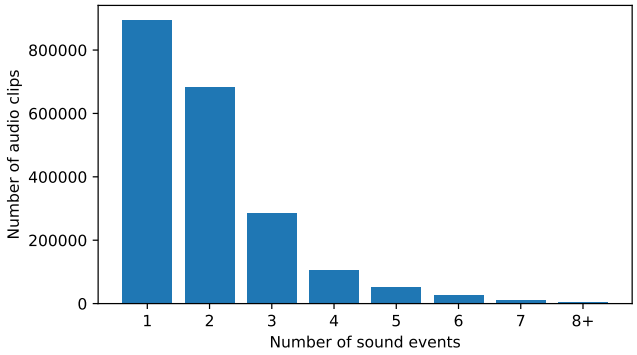


Fig. 4. Distribution of the number of sound classes in an audio clip.

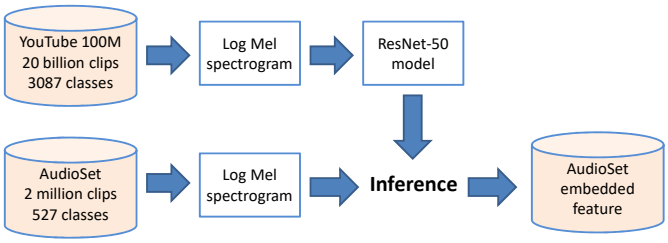


Fig. 5. A ResNet-50 model is trained on the YouTube 100M dataset. Audio clips from AudioSet are given as input to the trained ResNet-50 model to extract the bottleneck features, which are released by AudioSet.

The exponential function is defined as

$$\phi(z) = e^z. \tag{26}$$

The sigmoid function is defined as

$$\phi(z) = \frac{1}{1 + e^{-z}}. \tag{27}$$

For a vector $\mathbf{z}$, the softmax function is defined as

$$\phi(z_j) = \frac{e^{z_j}}{\sum_k e^{z_k}}. \tag{28}$$

The network-in-network function [55] is defined as

$$\phi(\mathbf{z}) = \sigma(\mathbf{H}_2(\mathbf{H}_1\mathbf{z} + \mathbf{d}_1) + \mathbf{d}_2), \tag{29}$$

where $\mathbf{H}_1$, $\mathbf{H}_2$ are transformation matrices, $\mathbf{d}_1$ and $\mathbf{d}_2$ are biases and $\sigma$ is the sigmoid function.

## VI. Experiments

### A. Dataset

We evaluate the proposed attention neural networks on AudioSet [9], which consists of 2,084,320 10-second audio clips extracted from YouTube video with a hierarchical ontology of 527 classes in the released version (v1). We released both Keras and PyTorch implementations of our code online[2]. AudioSet consists of a variety of sounds. AudioSet is multi-labelled, such that each audio clip may contain more than one sound class. Fig. 4 shows the statistics of the number of sound classes in the audio clips. All audio clips contain at least one label. Out of over 2,084,320 audio clips, there are 896,045 audio clips containing one sound class, followed by around 684,166 audio clips containing two sound classes. Only 4,661 audio clips have more than 7 labels.

[2]https://github.com/qiuqiangkong/audioset_classification

TABLE I
Baseline results of segment based method, IS and ES methods

| | mAP | AUC | d-prime |
|---|---|---|---|
| Random guess | 0.005 | 0.500 | 0.000 |
| Google baseline [9] | 0.314 | 0.959 | 2.452 |
| Segment based [13] | 0.293 | 0.960 | 2.483 |
| (IS) SMI assumption [11] | 0.292 | 0.960 | 2.471 |
| (IS) Collective assumption | 0.300 | 0.964 | 2.536 |
| (ES) Average instances [50] | **0.317** | **0.963** | **2.529** |
| (ES) Max instance | 0.284 | 0.958 | 2.443 |
| (ES) Min instance | 0.281 | 0.956 | 2.413 |
| (ES) Max-min instance [51] | 0.306 | 0.962 | 2.505 |

Instead of providing raw audio waveforms, AudioSet provides bottleneck features of audio clips. The bottleneck features are extracted from the bottleneck layer of a ResNet convolutional neural network, pre-trained on 70 million audio clips from the YouTube100M dataset [14]. To begin with, the 70 million training audio clips are segmented to non-overlapping 960 ms segments. Each segment inherits all tags of its parent video. Then short-time Fourier transform (STFT) is applied on each 960 ms segment with a window size of 25 ms and a hop size of 10 ms to obtain a spectrogram. Then a mel filter bank with 64 frequency bins is applied on the spectrograms followed by a logarithmic operation to obtain log mel spectrograms. Each log mel spectrogram of a segment has a shape of $96 \times 64$, representing the time steps and the number of mel frequency bins. A ResNet-50 neural network is trained on these log mel spectrograms with the 3087 most frequent labels. After training, the ResNet-50 is used as a feature extractor. By inputting an audio clip to the ResNet-50 network, the outputs of the bottleneck layer are used as bottleneck features of the audio clip. The framework of AudioSet feature extraction is shown in Fig. 5.

### B. Evaluation criterion

We first introduce basic statistics [56]: true positive (TP), where both the reference and the system prediction indicate an event to be active; false negative (FN), where the reference indicates an event is active but the system prediction indicates an event is inactive; false positive (FP), where the system prediction indicates an event is active but the reference indicates it is inactive; true negative (TN), where both the reference and the system prediction indicate an event is inactive. Precision (P) and recall (R) are defined as in [56]:

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \qquad \mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{30}$$

In addition, the false positive rate is defined as [56]

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}. \tag{31}$$

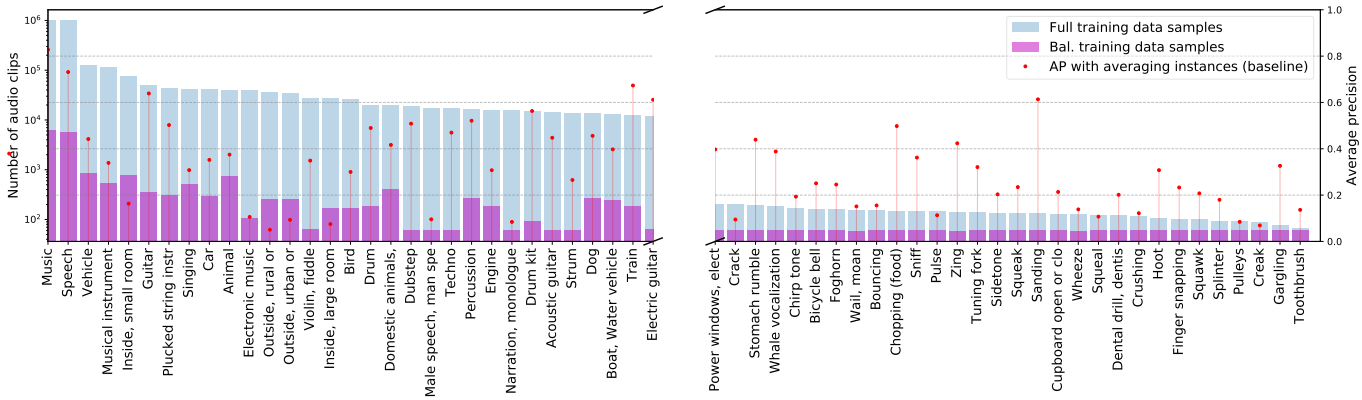


Fig. 6. AudioSet statistics. Upper bars: the number of audio clips of a specific sound class sorted in descending order plotted in log scale with respect to the sound classes. Red stems: average precision (AP) of sound classes with the feature-level attention model.

TABLE II
RESULTS OF ES AVERAGE INSTANCES METHOD WITH DIFFERENT BALANCING STRATEGY.

| | mAP | AUC | d-prime |
|---|---|---|---|
| Balanced data | 0.274 | 0.949 | 2.316 |
| Full data (no bal. training) | 0.268 | 0.950 | 2.331 |
| Full data (bal. training) | **0.317** | **0.963** | **2.529** |

Following [9], we adopt mean average precision (mAP), area under the curve (AUC) and d-prime as evaluation metrics. Average precision (AP) [9] is defined as the area under the recall-precision curve of a specific class. The mean average precision (mAP) is the average value of AP over all classes. As AP is regardless of TN, AUC is used as a complementary metric. AUC is the area under the receiver operating characteristic (ROC) created by plotting the recall against the false positive rate (FPR) at various threshold settings for a specific class. We use mAUC to denote the average value of AUC over all classes. R2: D-prime is a statistic used in signal detection theory that provides separation between signal and noise distributions. D-prime is obtained via a transformation of AUC and has a better dynamic range than AUC when AUC is larger than 0.9. A higher mAP, AUC and d-prime indicates a better performance. D-prime can be calculated by [9]:

$$\text{d-prime} = \sqrt{2} F_x^{-1}(\text{AUC}), \tag{32}$$

where $F_x^{-1}$ is an inverse of the cumulative distribution function defined by

$$F_x(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2}} dx. \tag{8}$$

### C. Baseline system

We build baseline systems with segment based method, IS and ES models without the attention mechanism described in Section III-A, IV-A and IV-C, respectively. In the segment based model, a classifier is trained on individual instances, where each instance inherits the tags of a bag. A three-layer fully-connected neural network with 1024 hidden units and ReLU [54] non-linearity is applied. Dropout [57] with a rate of 0.5 is used to prevent overfitting. The loss function for training is given in (1). In inference, the prediction is obtained by averaging the prediction of individual instances. The IS models have the same structure as the segment based model. Different from the segment based model, the instance-level predictions by the IS models are aggregated to a bag-level prediction by either the SMI assumption in (9) or CA in (10). The loss function is calculated from (8). The ES method aggregates the instances of a bag to an embedded vector before tagging. The embedding function can be the averaging mapping in (14) or max-min vector mapping in (15). Then the embedded vector is input to a neural network in the same way as the segment based model. The loss function is calculated from (8). We adopt the Adam optimiser [58] with a learning rate of 0.001 in training. The mini-batch size is set to 500. The networks are trained for a total number of 50,000 iterations. We average the predictions of 9 models from 10,000 to 50,000 iterations as the final prediction to ensemble and stabilise the result, which can reduce the prediction randomness caused by the model.

Table I shows the tagging result of segment based method, IS and ES baseline methods. The first row shows that the random guess achieves an mAP of 0.005, an AUC of 0.500 and a d-prime of 0. The segment based model achieves an mAP of 0.293, slightly better than the IS methods with the CA and SMI assumption, with mAP of 0.281 and 0.273, respectively. The fifth to the eighth rows show that both the ES methods with averaging and the max-min instances perform better than the segment based model and IS methods. Averaging the instances performs the best in the ES methods with an mAP of 0.317, an AUC of 0.963 and a d-prime of 2.529.

### D. Data balancing

AudioSet is highly imbalanced, as some sound classes such as speech and music are more frequent than others. The upper bars in Fig. 6 show the number of audio clips per class sorted in descending order (in log scale). The data has a long tail distribution. Music and speech appear in almost 1 million audio clips while some sounds such as gargling and toothbrush only appear in hundreds of audio clips. AudioSet provides a balanced

TABLE III
CORRELATION OF MAP WITH TRAINING EXAMPLES AND LABELLING ACCURACY OF SOUND CLASSES.

| | PCC | p-value |
|---|---|---|
| Training examples | 0.169 | $9.35 \times 10^{-5}$ |
| Labels quality | 0.230 | $7 \times 10^{-7}$ |

subset consisting of 22,160 audio clips. The lower bars in Fig. 6 show the number of audio clips per class of the balanced subset. When training a neural network, data is loaded in mini batches. We found that without a balancing strategy, the classes with fewer examples are less likely to be selected in training. Several balancing strategies have been investigated in image classification such as balancing the frequent and infrequent classes [59]. In this paper, we follow the mini-batch balancing strategy [15] for AudioSet tagging, where each mini-batch is balanced to have approximately the same number of examples in training the neural network.

We first investigate the performance of training on the balanced subset only and training on the full data. We adopt the best baseline model; that is, the ES average instances model in Section VI-C. Table II shows that the model trained with only the balanced subset achieves an mAP of 0.274. The model trained with the full dataset without balancing achieves an mAP of 0.268. The model trained with the balancing strategy achieves an mAP of 0.317. Fig. 7 shows the class-wise AP. The dashed and solid curves show the training and testing AP, respectively. In addition, Fig. 7 shows that the AP is not always positive related to the number of training examples. For example, when using full data for training, "bagpipes" has 1,715 audio clips but achieves an mAP of 0.884, while "outside" has 34,117 audio clips but only achieves an AP of 0.093. We discover that for a majority of sound classes, the improvement of AP is small compared when using the full dataset rather than the balanced subset. For example, there are 60 and 1,715 "bagpipes" audio clips in the balanced subset and the full dataset, respectively. Their APs are 0.873 and 0.884, respectively, indicating that collecting more data for "bagpipes" does not substantially improve its tagging result.

To investigate how AP is related to the number of training examples, we calculate their Pearson correlation efficient (PCC)[3]. PCC is a number between -1 and +1. The PCC of -1, 0, +1 indicate negative correlation, no correlation and positive correlation, respectively. R2: The null hypothesis is that the correlation of the pair of random variables is 0. The p-value indicates the confidence when the null hypothesis is satisfied. If the p-value is lower than the conventional 0.05 the PCC is called statistically significant. Table III shows that AP and the number of training examples have a correlation with a PCC of 0.169 and the p-value is $9.35 \times 10^{-5}$, indicating that AP is only weakly positively related with the number of training samples.

[3]Given a pair of random variables $X$ and $Y$, the PCC is calculated as $\frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y}$, where $\text{cov}(\cdot, \cdot)$ is the covariance of two variables and $\sigma$ is the standard deviation of the random variables.

### E. Noisy labels

AudioSet contains noisy tags [9]. That is, some tags for training may be incorrect. There are three major reasons leading to the noisy tags in AudioSet [9]: 1) confusing labels, where some sound classes are easily confused with others; 2) human error, where the labeling procedure may be flawed; 3) faint/non-salient sounds, where some sound are faint to recognise in an audio clip. Sound classes with a high label confidence include "christmas music" and "accordion". Sound classes with a low label confidence include "boiling" and "bicycle". To investigate how accurate are the ground truth tags, The authors of AudioSet conducted an internal quality assessment task where experts checked 10 random segments for most of the classes. R2: The quality is a value between 0 and 1 measured by the percentage of correctly labelled audio clips verified by human. The quality of labels is shown in Fig. 7 with red triangles. Hyphen symbols are plotted for the classes that have not been evaluated. We discover that AP is not always correlated positively with the quality of labels. For example, our model achieves an AP of 0.754 in recognizing "harpsichord", while the human label quality is 0.4. On the other hand, humans achieve a label quality of 1.0 in "hiccup", but the AP of our model is 0.076. Table III shows that AP and the quality of labels have a weak PCC of 0.230, indicating AP is only weakly correlated with the quality of labels.

### F. Attention neural networks

We evaluate the decision-level and the feature-level attention neural networks in this subsection. We adopt the architecture in Fig. 2(c) as our model. The output $\mathbf{x}'$ of the layer before the attention function and classifier bifurcate is obtained by (17). Then the decision-level and feature-level attention neural networks are modelled by (18, 19) and (23, 24), respectively. The first row of Table IV shows that the ES method with averaged instances achieves an mAP of 0.317. The second and third rows show that the JDC model in Fig. 2(a) and the self-attention model in Fig. 2(b) achieve an mAP of 0.337 and 0.324, respectively. The fourth and fifth row show that the decision-level attention neural network achieves an mAP of 0.337. The decision-level multiple attention neural network further improves this result to an mAP of 0.357.

The results of the feature-level attention neural networks are shown in the bottom block of Table IV. The ES methods with average and maximum aggregation achieve an mAP of 0.298 and 0.343, respectively. The feature-level attention neural network achieves an mAP of 0.361, an mAUC of 0.969 and a d-prime of 2.641, outperforming the other models. One explanation is that the feature-level attention neural network can attend to or ignore the features in the feature space which further improves the capacity of the decision-level attention neural network. Fig. 8 shows the class-wise performance of the attention neural networks. The feature-level attention neural network outperforms the decision-level attention neural network and the ES method with averaged instances in a majority of sound classes. The results of all 527 sound classes are shown in Fig. 9 in the Appendix.

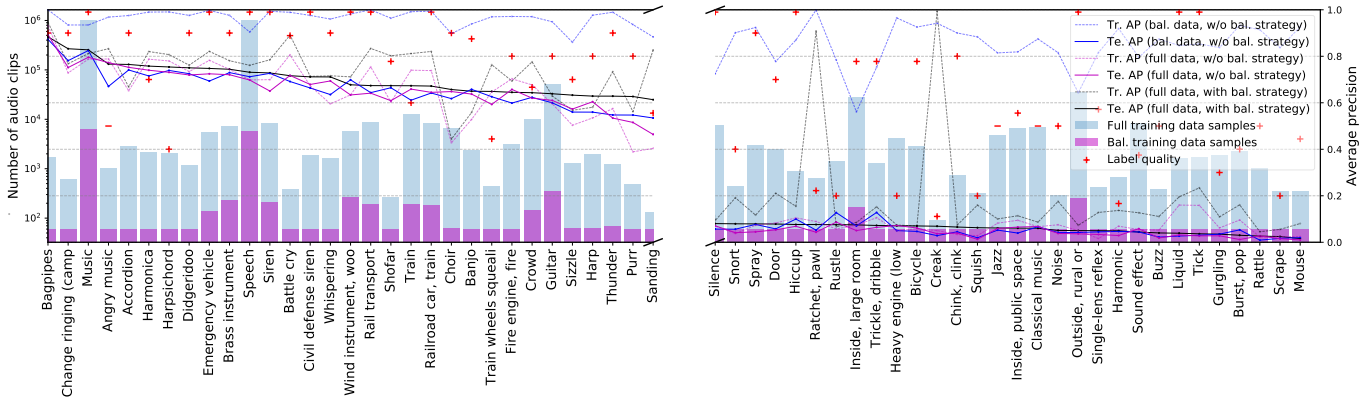


Fig. 7. Class-wise AP of sound events using the IS average instances model trained with different balancing strategy. Abbreviations: Tr.: Training; Te.: Testing; bal.: training with balanced subset; full: trained with full dataset; w/o: without mini-batch data balancing; w.: with mini-batch data balancing.

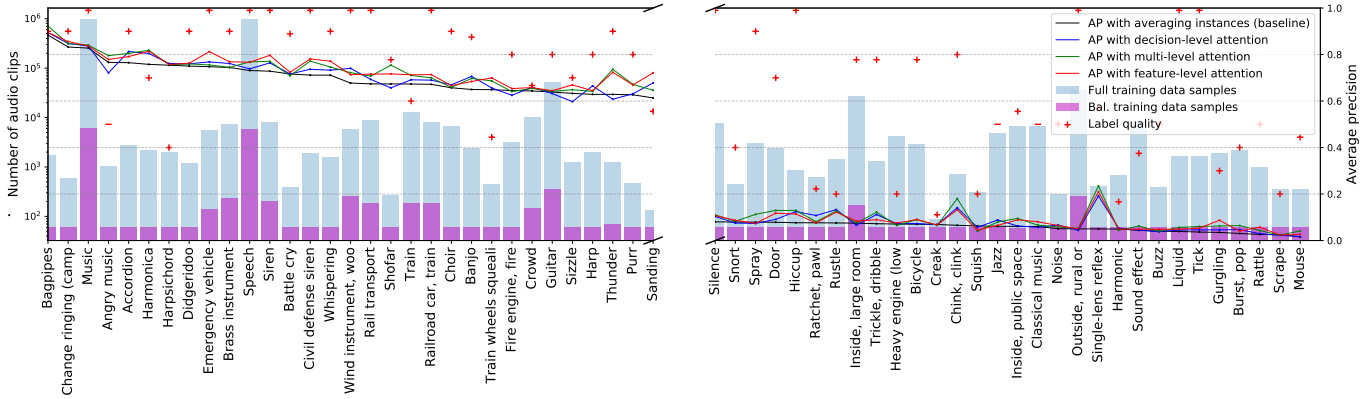


Fig. 8. Class-wise AP of sound events predicted using different models.

TABLE IV
RESULTS OF DECISION-LEVEL ATTENTION MODEL AND FEATURE-LEVEL ATTENTION MODEL

| | mAP | AUC | d-prime |
|---|---|---|---|
| Average instances [50] | 0.317 | 0.963 | 2.529 |
| JDC [12] | 0.337 | 0.963 | 2.526 |
| Self attention [48] | 0.324 | 0.962 | 2.506 |
| Decision-level single-attention [15] | 0.337 | 0.968 | 2.612 |
| Decision-level multi-attention [16] | 0.357 | 0.968 | 2.621 |
| Feature-level avg. pooling | 0.298 | 0.960 | 2.475 |
| Feature-level max pooling | 0.343 | 0.966 | 2.589 |
| Feature-level attention | **0.361** | **0.969** | **2.641** |

TABLE V
RESULTS OF MODELING THE NON-NEGATIVE $\phi_2$ WITH DIFFERENT NON-NEGATIVE FUNCTIONS.

| | mAP | AUC | d-prime |
|---|---|---|---|
| ReLU att | 0.308 | 0.963 | 2.520 |
| Exp. att | 0.358 | 0.969 | 2.631 |
| Sigmoid att | **0.361** | **0.969** | **2.641** |
| Softmax att | 0.360 | 0.969 | 2.636 |
| NIN | 0.359 | 0.969 | 2.637 |

### G. Modeling attention function with different functions

As described in Section V-C, we model the attention function $q$ of the feature-level attention neural network via a non-negative function $\phi_2$. The choice of the non-negative function may affect the optimisation and result of the attention neural network. Table V shows that the exponential, sigmoid, softmax and NIN functions achieve a similar mAP of approximately 0.360. Modeling $\phi(\cdot)$ with ReLU is worse than the others.

### H. Attention neural networks with different embedding depth and width

As shown in (17), our attention neural networks map the instances $\mathbf{x}$ to $\mathbf{x}'$ through several non-linear embedding layers to increase the representation ability of the instances. We model $f_{\text{FC}}$ using the feature-level attention neural network with fully-connected layers with different depths. Table VI shows that the mAP increases from 0 layers and reaches a peak of 0.361 at 3 layers. More hidden layers do not increase the mAP. The reason might be that the AudioSet bottleneck features obtained by a ResNet-50 trained on YouTube100M have good separability. Therefore, there is no need to apply very deep neural networks on the AudioSet bottleneck features. On the other hand, the YouTube100M data may have a different distribution from

TABLE VI
RESULTS OF MODELING THE ATTENTION NEURAL NETWORK WITH DIFFERENT LAYER DEPTHS.

| Depth | mAP | AUC | d-prime |
|---|---|---|---|
| 0 | 0.328 | 0.963 | 2.522 |
| 1 | 0.356 | 0.967 | 2.605 |
| 2 | 0.358 | 0.968 | 2.620 |
| 3 | **0.361** | **0.969** | **2.641** |
| 4 | 0.356 | 0.969 | 2.637 |
| 6 | 0.348 | 0.968 | 2.619 |
| 8 | 0.339 | 0.967 | 2.595 |
| 10 | 0.331 | 0.966 | 2.579 |

TABLE VII
RESULTS OF MODELING THE ATTENTION NEURAL NETWORK WITH DIFFERENT NUMBER OF HIDDEN UNITS.

| Hidden units | mAP | AUC | d-prime |
|---|---|---|---|
| 256 | 0.305 | 0.962 | 2.512 |
| 512 | 0.339 | 0.967 | 2.599 |
| 1024 | 0.361 | 0.969 | **2.641** |
| 2048 | **0.369** | **0.969** | 2.640 |
| 4096 | 0.369 | 0.968 | 2.619 |

AudioSet. As a result, the embedding mapping $f_{\text{FC}}$ can be used as domain adaption.

Based on the network $f_{\text{FC}}$ modelled with three layers in the feature-level attention neural network, we investigate the width of $f_{\text{FC}}$. Table VII shows that feature-level attention model with 2048 hidden units in each hidden layer achieves an mAP of 0.369, an mAUC of 0.969 and a d-prime of 2.641 is achieved, outperforming the models with 256, 512, 1024 and 4096 hidden units in each layer. On the other hand, with 4096 hidden units, the model tends to overfit, and does not outperform the model with 2048 hidden units.

## VII. CONCLUSION

We have presented a decision-level and a feature-level attention neural network for AudioSet tagging. We developed the connection between multiple instance learning and attention neural networks. We investigated the class-wise performance of all the 527 sound classes in AudioSet and discovered that the AudioSet tagging performance on AudioSet embedding features is only weakly correlated with the number of training examples and quality of labels, with Pearson correlation coefficients of 0.169 and 0.230, respectively. In addition, we investigated modelling the attention neural networks with different attention functions, depths and widths. Our proposed feature-level attention neural network achieves a state-of-the-art mean average precision (mAP) of 0.369 compared to the best MIL method of 0.317 and the decision-level attention neural network of 0.337. In the future, we will explore weakly labelled sound event detection on AudioSet with attention neural networks.

## ACKNOWLEDGMENT

The authors would like to thank all anonymous reviewers for their suggestions to improve this paper.

## REFERENCES

[1] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, "A survey of audio-based music classification and annotation," *IEEE Transactions on Multimedia*, vol. 13, pp. 303–319, 2011.

[2] R. Typke, F. Wiering, and R. C. Veltkamp, "A survey of music information retrieval systems," in *International Conference on Music Information Retrieval*, 2005, pp. 153–160.

[3] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: An IEEE AASP challenge," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2013.

[4] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[5] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: Outcome of the DCASE 2016 challenge," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 2, pp. 379–393, 2018.

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

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[8] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85 – 117, 2015.

[9] 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.

[10] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[11] A. Kumar and B. Raj, "Audio event detection using weakly labeled data," in *Proceedings of the 2016 ACM on Multimedia Conference*, 2016, pp. 1038–1047.

[12] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, "A joint detection-classification model for audio tagging of weakly labelled data," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 641–645.

[13] Q. Kong, I. Sobieraj, W. Wang, and M. D. Plumbley, "Deep neural network baseline for DCASE challenge 2016," in *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2016.

[14] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, "CNN architectures for large-scale audio classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 131–135.

[15] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, "Audio set classification with attention model: A probabilistic perspective," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 316–320.

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

[17] O. Maron and T. Lozano-Pérez, "A framework for multiple-instance learning," in *Advances in Neural Information Processing Systems (NIPS)*, 1998, pp. 570–576.

[18] 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 *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2015.

[19] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *ACM International Conference on Multimedia*. ACM, 2014, pp. 1041–1044.

[20] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[21] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *IEEE European Signal Processing Conference (EUSIPCO)*, 2016, pp. 1128–1132.

[22] Mesaros, A. and Heittola, T. and Virtanen, T., "A multi-device dataset for urban acoustic scene classification," in *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2018.
[23] E. Fonseca, M. Plakal, F. Font, D. P. W. Ellis, X. Favory, J. Pons, and X. Serra, "General-purpose tagging of freesound audio with audioset labels: Task description, dataset, and baseline," in *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2018.
[24] D. Stowell, M. D. Wood, H. Pamuła, Y. Stylianou, and H. Glotin, "Automatic acoustic detection of birds through deep learning: The first bird audio detection challenge," *Methods in Ecology and Evolution*, 2018.
[25] D. Li, I. K. Sethi, N. Dimitrova, and T. McGee, "Classification of general audio data for content-based retrieval," *Pattern Recognition Letters*, vol. 22, no. 5, pp. 533–544, 2001.
[26] B. Uzkent, B. D. Barkana, and H. Cevikalp, "Non-speech environmental sound classification using svms with a new set of features," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 5, pp. 3511–3524, 2012.
[27] H. Eghbal-Zadeh, B. Lehner, M. Dorfer, and G. Widmer, "CP-JKU submissions for DCASE-2016: a hybrid approach using binaural i-vectors and deep convolutional neural networks," DCASE2016 Challenge, Tech. Rep., 2016.
[28] J. Aucouturier, B. Defreville, and F. Pachet, "The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music," *The Journal of the Acoustical Society of America*, vol. 122, no. 2, pp. 881–891, 2007.
[29] S. Sigtia, A. M. Stark, S. Krstulović, and M. D. Plumbley, "Automatic environmental sound recognition: Performance versus computational cost," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 2096–2107, 2016.
[30] E. Cakır, T. Heittola, and T. Virtanen, "Domestic audio tagging with convolutional neural networks," DCASE2016 Challenge, Tech. Rep., 2016.
[31] K. Choi, G. Fazekas, and M. Sandler, "Automatic tagging using deep convolutional neural networks," in *Conference on International Society for Music Information Retrieval (ISMIR)*, 2016.
[32] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2392–2396.
[33] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Large-scale weakly supervised audio classification using gated convolutional neural network," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 121–125.
[34] S. Parekh, A. Ozerov, S. Essid, N. Duong, P. Pérez, and G. Richard, "Identify, locate and separate: Audio-visual object extraction in large video collections using weak supervision," *arXiv preprint arXiv:1811.04000*, 2018.
[35] J. Amores, "Multiple instance classification: Review, taxonomy and comparative study," *Artificial Intelligence*, vol. 201, pp. 81–105, 2013.
[36] S.-Y. Chou, J.-S. R. Jang, and Y.-H. Yang, "Learning to recognize transient sound events using attentional supervision." in *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2018, pp. 3336–3342.
[37] S. Pancoast and M. Akbacak, "Bag-of-audio-words approach for multimedia event classification," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.
[38] B. Sankaran, H. Mi, Y. Al-Onaizan, and A. Ittycheriah, "Temporal attention model for neural machine translation," *arXiv preprint arXiv:1608.02927*, 2016.
[39] M. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.
[40] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning (ICML)*, 2015, pp. 2048–2057.
[41] G. Liu, J. Yang, and Z. Li, "Content-based image retrieval using computational visual attention model," *Pattern Recognition*, vol. 48, no. 8, pp. 2554–2566, 2015.
[42] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, no. 1-2, pp. 31–71, 1997.
[43] Q. Zhang and S. A. Goldman, "EM-DD: An improved multiple-instance learning technique," in *Advances in Neural Information Processing Systems (NIPS)*, 2002, pp. 1073–1080.
[44] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2003, pp. 577–584.
[45] Z. Zhou and M. Zhang, "Neural networks for multi-instance learning," in *International Conference on Intelligent Information Technology (ICIIT)*, 2002, pp. 455–459.
[46] X. Wang, Y. Yan, P. Tang, X. Bai, and W. Liu, "Revisiting multiple instance neural networks," *Pattern Recognition*, vol. 74, pp. 15–24, 2018.
[47] Y. Wang, J. Li, and F. Metze, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 31–35.
[48] M. Ilse, J. M. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," in *International Conference on Machine Learning (ICML)*, 2018.
[49] J. Wang and J. Zucker, "Solving multiple-instance problem: A lazy learning approach," in *International Conference on Machine Learning (ICML)*, 2000.
[50] L. Dong, "A comparison of multi-instance learning algorithms," Ph.D. dissertation, The University of Waikato, 2006.
[51] T. Gärtner, P. A. Flach, A. Kowalczyk, and A. J. Smola, "Multi-instance kernels," in *International Conference on Machine Learning (ICML)*, vol. 2, 2002, pp. 179–186.
[52] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *International Conference on Learning Representations (ICLR)*, 2015.
[53] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," in *International Conference on Learning Representations (ICLR)*, 2017.
[54] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *International Conference on Machine Learning (ICML)*, 2010, pp. 807–814.
[55] M. Lin, Q. Chen, and S. Yan, "Network in network," *International Conference on Learning Representations (ICLR)*, 2014.
[56] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, 2016.
[57] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
[58] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2014.
[59] L. Shen, Z. Lin, and Q. Huang, "Relay backpropagation for effective learning of deep convolutional neural networks," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 467–482.

**Qiuqiang Kong** (S'17) received the B.Sc. and M.E. degrees from South China University of Technology, Guangzhou, China, in 2012 and 2015, respectively. He is currently working toward the Ph.D. degree from the University of Surrey, Guildford, U.K on sound event detection. His research topic includes sound understanding, audio signal processing and machine learning. He was nominated as the postgraduate research student of the year in University of Surrey, 2019.

**Changsong Yu** received the B.E. degree from Anhalt University of Applied Sciences and M.S. degree University of Stuttgart, Germany, in 2015 and 2018, respectively. He is currently working as simultaneous localization and mapping (SLAM) algorithm engineer in HoloMatic, Beijing, China. His research interest includes deep learning and SLAM.

**Yong Xu** (M'17) received the Ph.D. degree from the University of Science and Technology of China (USTC), Hefei, China, in 2015, on the topic of DNN-based speech enhancement and recognition. Currently, he is a senior research scientist in Tencent AI lab, Bellevue, USA. He once worked at the University of Surrey, U.K. as a Research Fellow from 2016 to 2018 working on sound event detection. He visited Prof. Chin-Hui Lee's lab in Georgia Institute of Technology, USA from Sept. 2014 to May 2015. He once also worked in IFLYTEK company from 2015 to 2016 to develop far-field ASR technologies. His research interests include deep learning, speech enhancement and recognition, sound event detection, etc. He received 2018 IEEE SPS best paper award.

**Turab Iqbal** received the B.Eng. degree in Electronic Engineering from the University of Surrey, U.K., in 2017. Currently, he is working towards a Ph.D. degree from the Centre for Vision, Speech and Signal Processing (CVSSP) in the University of Surrey. His research interests are mainly in machine learning using weakly labeled data for audio classification and localization.

**Wenwu Wang** (M'02-SM'11) was born in Anhui, China. He received the B.Sc. degree in 1997, the M.E. degree in 2000, and the Ph.D. degree in 2002, all from Harbin Engineering University, China. He then worked in King's College London, Cardiff University, Tao Group Ltd. (now Antix Labs Ltd.), and Creative Labs, before joining University of Surrey, UK, in May 2007, where he is currently a professor in signal processing and machine learning, and a Co-Director of the Machine Audition Lab within the Centre for Vision Speech and Signal Processing. He has been a Guest Professor at Qingdao University of Science and Technology, China, since 2018. His current research interests include blind signal processing, sparse signal processing, audio-visual signal processing, machine learning and perception, machine audition (listening), and statistical anomaly detection. He has (co)-authored over 200 publications in these areas. He served as an Associate Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING from 2014 to 2018. He is also Publication Co-Chair for ICASSP 2019, Brighton, UK.

**Mark D. Plumbley** (S'88-M'90-SM'12-F'15) received the B.A.(Hons.) degree in electrical sciences and the Ph.D. degree in neural networks from University of Cambridge, Cambridge, U.K., in 1984 and 1991, respectively. Following his PhD, he became a Lecturer at King's College London, before moving to Queen Mary University of London in 2002. He subsequently became Professor and Director of the Centre for Digital Music, before joining the University of Surrey in 2015 as Professor of Signal Processing. He is known for his work on analysis and processing of audio and music, using a wide range of signal processing techniques, including matrix factorization, sparse representations, and deep learning. He is a co-editor of the recent book on Computational Analysis of Sound Scenes and Events, and Co-Chair of the recent DCASE 2018 Workshop on Detection and Classifications of Acoustic Scenes and Events. He is a Member of the IEEE Signal Processing Society Technical Committee on Signal Processing Theory and Methods, and a Fellow of the IET and IEEE.

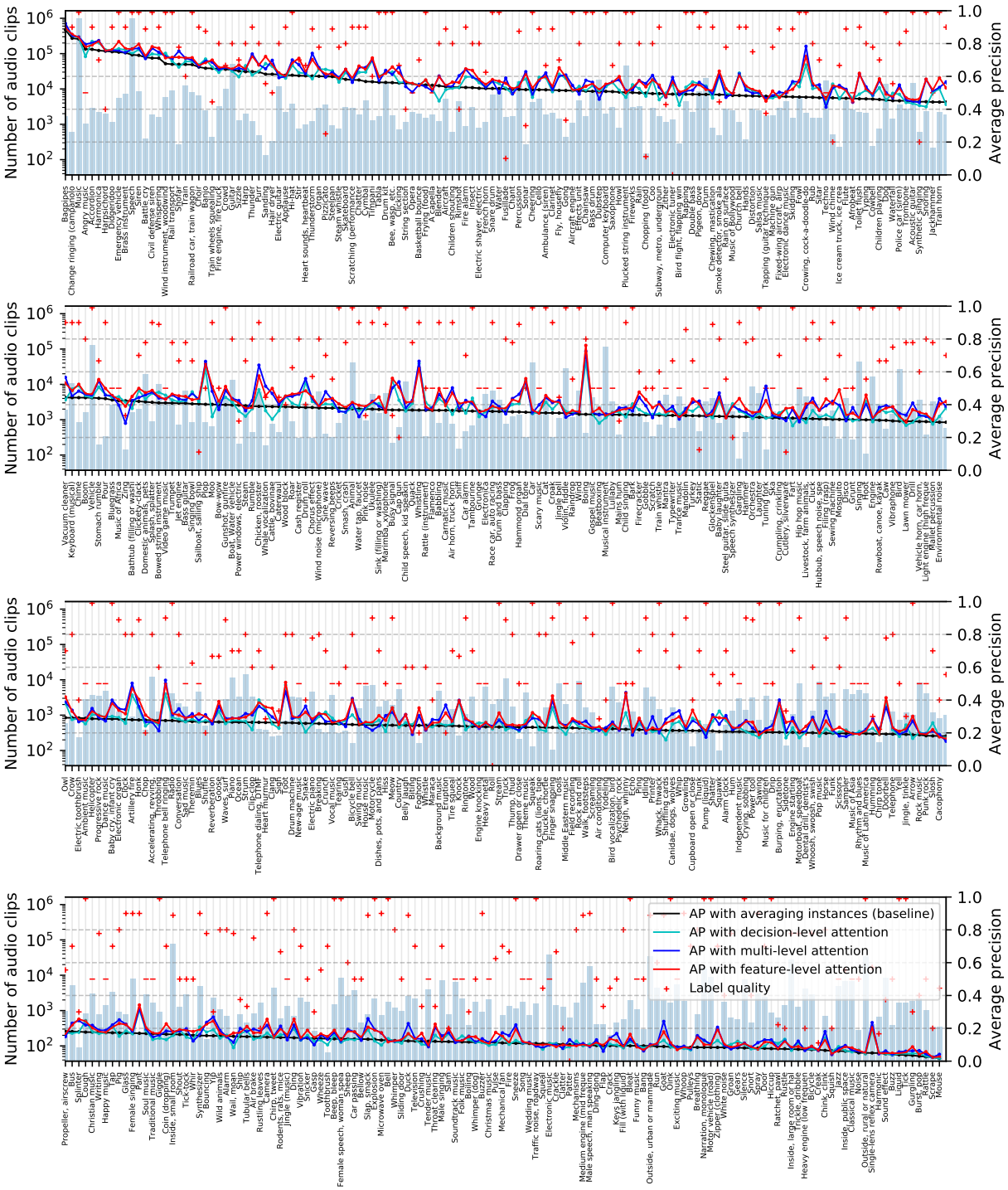


Fig. 9. mAP of all sound classes predicted using different models.