# Modeling Label Dependencies for Audio Tagging With Graph Convolutional Network

Helin Wang, Yuexian Zou, *Senior Member, IEEE*, Dading Chong, and Wenwu Wang, *Senior Member, IEEE*

***Abstract*—As a multi-label classification task, audio tagging aims to predict the presence or absence of certain sound events in an audio recording. Existing works in audio tagging do not explicitly consider the probabilities of the co-occurrences between sound events, which is termed as the label dependencies in this study. To address this issue, we propose to model the label dependencies via a graph-based method, where each node of the graph represents a label. An adjacency matrix is constructed by mining the statistical relations between labels to represent the graph structure information, and a graph convolutional network (GCN) is employed to learn node representations by propagating information between neighboring nodes based on the adjacency matrix, which implicitly models the label dependencies. The generated node representations are then applied to the acoustic representations for classification. Experiments on Audioset show that our method achieves a state-of-the-art mean average precision (mAP) of 0.434.**

***Index Terms*—Audio tagging, label dependencies, graph convolutional network, representation learning.**

## I. Introduction

AUDIO tagging [1] is the task of predicting the presence or absence of sound events within an audio clip, which has many applications such as information retrieval [2] and music tagging [3]. Compared to single-label audio classification [4], [5], one of the challenges in audio tagging is to deal with the multiple labels in an audio recording.

Recently, convolutional neural networks (CNNs) [6]–[11] and convolutional recurrent neural networks (CRNNs) [12]–[14] provide the state-of-the-art results in audio tagging tasks, which show powerful ability to learn acoustic representations from manually-design features, such as log mel spectrograms. In most previous methods, each sound event type is considered independently, so that audio tagging is treated as a binary classification problem for each sound event type. As a result, the intrinsic relationships between sound events are ignored in these methods. As sound events often co-occur in an audio clip, (*e.g.* when the sound event *piano* appears, *guitar* is more likely to appear than *babycry*), it would be beneficial to take into account the dependencies among labels. Meanwhile, labels often conform to the ontology structure of the abstract sound categories [15]. For example, *snake* can be categorized either as a general category of *animal* or a more specific category of *wild animal*. Some approaches have been proposed to capture the relationships among labels for audio classification. In [16], graph Laplacian regularization was introduced to model the co-occurrence of sound events, and Xu *et al.* [17] proposed a deep neural network (DNN)-based hierarchical learning method for acoustic scene classification. In addition, SONYC Urban Sound Tagging (SONYC-UST) [18] containing 8 coarse-grained classes and 23 fine-grained classes was presented for the DCASE 2019 Urban Sound Tagging Challenge [19]. For large-scale multi-label datasets (*e.g.* Audioset [21]), which contain numerous categories, the hierarchical structures are not clearly pre-defined. However, the implicit label dependencies could be explored to achieve better classification performance.

In this letter, we model the label dependencies via a graph, which has been proven to be effective in capturing the relationships among labels [22]–[24]. The main contribution of this letter is that the implicit dependencies between labels are modeled via GCN with the statistical relations between labels. This is different from two contemporary works [15], [20] brought to our attention where the ontology based domain knowledge is used for the graph construction, rather than the statistical relations exploited in our work. More specifically, each edge in the graph represents the relationships between two nodes, and the adjacency matrix is constructed by the conditional probabilities between labels within the dataset to represent the graph structure information. GCN is employed to learn node representations using the graph structure information, which are then applied to the acoustic representations as the label-wise weights for classification. A single layer GCN learns the representations of each node by aggregating the information of its immediate neighbors. While in multi-layer GCN, information is propagated from more neighbors, hence implicitly modeling the label dependencies. In addition, re-weighting schemes are proposed to alleviate the over-fitting and over-smoothing problems for the adjacency matrix.

## II. Graph Convolutional Network

Graph convolutional network (GCN) was presented for semi-supervised learning on graph-structured data [25]. The main

Manuscript received May 16, 2020; revised July 10, 2020; accepted August 16, 2020. Date of publication August 26, 2020; date of current version September 15, 2020. This work was supported in part by the Shenzhen Science and Technology Fundamental Research Programs under Grant JCYJ20170817160058246 and Grant JCYJ20180507182908274 and in part by the Collaboration Research Project funded by PKU-HKUST ShenZhen-HongKong Institution. The associate editor coordinating the review of this manuscript and approving it for publication was Mr. Ville M. Hautamaki. *(Corresponding author: Yuexian Zou.)*

Helin Wang, Yuexian Zou, and Dading Chong are with the School of Electronic, and Computer Engineering, Peking University, Shenzhen 518055, China (e-mail: wanghl15@pku.edu.cn; zouyx@pku.edu.cn; 1601213984@pku.edu.cn).

Wenwu Wang is with the Centre for Vision, Speech, and Signal Processing, University of Surrey, Surrey GU27XH, U.K. (e-mail: w.wang@surrey.ac.uk).

Digital Object Identifier 10.1109/LSP.2020.3019702

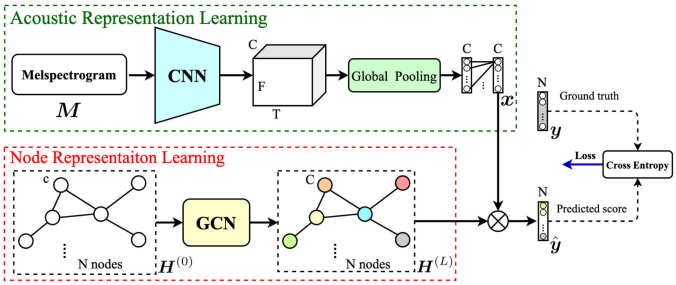



Fig. 1. Overall architecture of our AT-GCN model. Node representations are initially obtained by the word embeddings of the labels and the final node representations learned by GCN are applied to the acoustic representations for classification.

idea of GCN is to learn the node representations by aggregating the information of neighboring nodes on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $n$ nodes $v_i \in \mathcal{V}$, edges $(v_i, v_j) \in \mathcal{E}$. Let $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ be the adjacency matrix of $\mathcal{G}$, each GCN layer takes the node representations $\boldsymbol{H}^{(l)} \in \mathbb{R}^{n \times c}$ from the previous layer as inputs and outputs updated node representations $\boldsymbol{H}^{(l+1)} \in \mathbb{R}^{n \times c'}$, where $c$ and $c'$ indicate the dimensions of node features in the $l$-th layer and the $(l+1)$-th layer, respectively. A multi-layer GCN follows the layer-wise propagation rule between the nodes [25]:

$$\boldsymbol{H}^{(l+1)} = h\left(\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}\right) \tag{1}$$

Here, $\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}$ is the normalized symmetric adjacency matrix. $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}_N$ is the adjacency matrix with added self-connections ($\boldsymbol{I}_N$ is the identity matrix) and $\tilde{\boldsymbol{D}}$ is the diagonal degree matrix, where $\tilde{D_{ii}} = \sum_j \tilde{A}_{ij}$. $\boldsymbol{W}^{(l)} \in \mathbb{R}^{c \times c'}$ is a trainable weight matrix, and $h(\cdot)$ denotes an activation function.

## III. Proposed Method

In this section, we present a graph-based method to model the label dependencies for audio tagging. The graph structure is constructed by the statistical relations between the labels, and GCN is employed to learn node representations on the graph. The generated node representations are then applied to the acoustic representations for classification, as detailed next.

### A. GCN for Audio Tagging

The overall architecture of our proposed model (AT-GCN) is shown in Fig. 1, and CNN10 [11] is used as the baseline model in our experiments. See [11] for details about CNN10.

**Acoustic representation learning** The aim of acoustic representation learning module is to extract the acoustic feature from the input log mel spectrogram, and our proposed AT-GCN has the same acoustic representation learning module as CNN10 [11]. Specifically, convolutional layers are applied to the spectrogram $\boldsymbol{M} \in \mathbb{R}^{t \times f}$, followed by a global pooling layer and a fully-connected layer. Following [11], both maximum and average operations are used for global pooling. Let $f_{\text{cnn}}, f_{\text{gp}}, f_{\text{fc}}$ be the operations of the convolutional layers, the global pooling layer and the fully-connected layer, respectively. The acoustic feature $\boldsymbol{x} \in \mathbb{R}^C$ (where $C$ denotes the dimensionality of the acoustic feature) can be obtained by

$$\boldsymbol{x} = f_{\text{fc}}\left(f_{\text{gp}}\left(f_{\text{cnn}}\left(\boldsymbol{M}; \theta_{\text{cnn}}\right)\right); \theta_{\text{fc}}\right) \tag{2}$$

Here, $\theta_{\text{cnn}}$ and $\theta_{\text{fc}}$ denote the model parameters of the convolutional layers and the fully-connected layer, respectively.

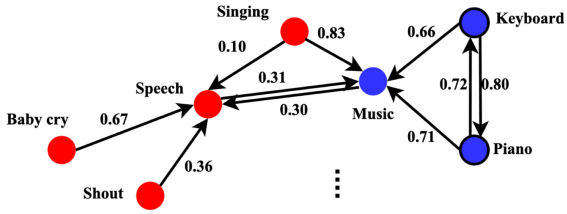



Fig. 2. An example of the graph to model the label dependencies. Each node represents a label and each edge represents the relationships between two nodes, which is determined by the conditional probabilities. Note that the edges with small values of the probabilities are filtered by a re-weighting scheme and not shown in the figure.

**Node representation learning** GCN is employed to learn node representations in our method. As in (1), the stacked multiple GCN layers are applied where each GCN layer takes the node representations $\boldsymbol{H}^{(l)}$ from the previous layer as inputs and outputs new node representations $\boldsymbol{H}^{(l+1)}$. The input node representations $\boldsymbol{H}^{(0)} \in \mathbb{R}^{N \times c}$ of the first GCN layer are the word embeddings of the labels, where $N$ denotes the number of labels and $c$ is the dimensionality of the embeddings. For the last layer (assuming that the number of layers is $L$), the output is $\boldsymbol{H}^{(L)} \in \mathbb{R}^{N \times C}$, where $C$ equals the dimensionality of the acoustic feature. The predicted score $\hat{\boldsymbol{y}} \in \mathbb{R}^N$ is then obtained by applying the last node representations to the acoustic representations [28].

$$\hat{\boldsymbol{y}} = \sigma\left(\boldsymbol{H}^{(L)}\boldsymbol{x}\right) \tag{3}$$

where $\sigma(\cdot)$ is the sigmoid function to restrict $\hat{y}^{(i)} \in (0, 1)$. For the given ground truth of the labels within an audio clip $\boldsymbol{y} \in \mathbb{R}^N$ (where $y^{(i)} = \{0, 1\}$ denotes whether label $i$ appears or not), the loss $\mathcal{L}$ is calculated using binary cross-entropy:

$$\mathcal{L} = \sum_{i=1}^{N} y^{(i)} \log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right) \tag{4}$$

### B. Construction of the Graph Structure

The graph structure determines the information propagation between nodes, however, there is no pre-defined graph structure in any audio tagging datasets. In our work, an adjacency matrix is constructed via mining the conditional probabilities between labels within the dataset to represent the graph structure information, as shown in Fig. 2.

Firstly, we count the occurrence of label pairs in the training set and get the matrix $\boldsymbol{X} \in \mathbb{R}^{N \times N}$ (where $N$ indicates the number of labels, and $X_{ij}$ denotes the co-occurring times of label $L_i$ and $L_j$). Then, the occurrence times of the labels in the training set are counted and the conditional probability matrix can be calculated by

$$P_{ij} = X_{ij}/T_i \tag{5}$$

**Algorithm 1:** AT-GCN.

**Input:** The log mel spectrogram $\boldsymbol{M}$;
The initial node representations $\boldsymbol{H}^{(0)}$;
**Output:** The predicted score $\hat{\boldsymbol{y}}$;
Calculate the adjacency matrix $\boldsymbol{A}$ using equation (5)-(7);
Extract the acoustic representation $\boldsymbol{x}$ using equation (2);
**for** $l = 0, \ldots, L-1$ **do**
    Get the node representations of the next layer $\boldsymbol{H}^{(l+1)}$ using equation (1);
**end for**
Calculate $\hat{\boldsymbol{y}}$ according to equation (3);
**return** $\hat{\boldsymbol{y}}$;

where $T_i$ denotes the occurrence times of label $L_i$ in the training set, $P_{ij} = P(L_j|L_i)$ means the probability of label $L_j$ when label $L_i$ appears. Note that $P_{ij}$ is not equal to $P_{ji}$ since $T_i$ is not the same as $T_j$.

However, the conditional probabilities between labels in the training and test set may not be completely consistent, and some small probabilities may become noise. Thus, it is necessary to alleviate over-fitting of the adjacency matrix. Specifically, a threshold $\tau$ is applied to filter noisy edges:

$$A_{ij} = \begin{cases} 0, & \text{if } P_{ij} < \tau \\ P_{ij}, & \text{if } P_{ij} \geq \tau \end{cases} \tag{6}$$

where $\boldsymbol{A}$ is the re-weighted adjacency matrix.

Another potential problem is over-smoothing, *i.e.* as the GCN layer deepens, the node features may be over-smoothed and nodes from different clusters may become indistinguishable [26], [27]. Thus, we adjust the information propagation among nodes in GCN by another re-weighting scheme:

$$A'_{ij} = \begin{cases} pA_{ij} / \sum_{\substack{j=1 \\ i \neq j}}^{N} A_{ij}, & \text{if } i \neq j \\ 1 - p, & \text{if } i = j \end{cases} \tag{7}$$

where $\boldsymbol{A}'$ is the re-weighted and normalized adjacency matrix, and $p$ determines the weights assigned to a node itself and its neighbors. When $p \to 0$, the neighboring information tends to be ignored. On the contrary, when $p \to 1$, the information of a node itself will not be considered.

The proposed algorithm is summarized in Algorithm 1.

## IV. Experiments

A large-scale multi-label dataset (Audioset [21]) is used in our experiments to evaluate our method, which is one of the most challenging datasets for audio tagging [1]. Log mel spectrograms are extracted from the audio signals as the input of the networks. The details are as follows.

### A. Dataset, Metrics and Preprocessing

**Dataset** Audioset [21] is a large-scale dataset with over 2 million 10-second audio clips from YouTube videos, with a total of 527 categories. The training set consists of 2,063,839 audio clips including a balanced subset of 22,160 audio clips. The evaluation set consists of 20,371 audio clips. Following [14], both the balanced and unbalanced training sets are used for training, with one part taken as our validation set. The evaluation set is used as the test set in our experiments.

**Metrics** Mean average precision (mAP), mean area under the curve (mAUC) and d-prime are used as our evaluation metrics. These metrics are calculated on individual classes and then averaged across all classes.

**Preprocessing** Limited by the computation resource, the pre-extracted log mel spectrograms [14] with window size 50 ms and hop length 25 ms are used in our experiments instead of the raw audio signals, which have lower time domain resolution than those used in [11]. The number of Mel bands is set to 64 and the size of a log mel spectrogram is $400 \times 64$.

TABLE I
Comparison of Performance on Audioset

| Model | Depth | mAP | mAUC | d-prime |
|---|---|---|---|---|
| Google CNN (2017) [21] | - | 0.314 | 0.959 | 2.452 |
| Multi-level attention (2018) [9] | - | 0.360 | 0.970 | 2.660 |
| Multi-level attention* (2018) [9] | - | 0.362 | 0.970 | 2.667 |
| TAL Net (2019) [14] | - | 0.362 | 0.965 | 2.554 |
| TAL Net* (2019) [14] | - | 0.367 | 0.969 | 2.638 |
| DeepRes (2019) [5] | - | 0.392 | 0.971 | 2.682 |
| CNN10* (2019) [11] | - | 0.422 | 0.970 | 2.653 |
| **AT-GCN (ours)** | 1-layer | 0.428 | 0.971 | 2.711 |
| | 2-layers | **0.434** | **0.974** | **2.736** |
| | 3-layers | 0.430 | 0.972 | 2.715 |

*The listed results of Multi-level attention [9], TAL Net [14] and CNN10 [11] are obtained under the same experimental setups as AT-GCN (e.g. using preprocessing and data augmentation). The original Multi-level attention [9] and TAL Net [14] did not use data augmentation.

### B. Implementation Details

**AT-GCN** The node representation learning module of our AT-GCN [1] consists of two GCN layers with output dimensionality of 256 and 512. It was proven that the performance is hardly impacted by the different initial label representations [28], and following [28], the word embeddings of 300-dim GloVe[2] [29] are used as the initial label representations in our experiments. The dimensionality of the acoustic representation is set to 512 for a fair comparison with CNN10 [11], and the hyperparameters $\tau$ in (6) and $p$ in (7) are set to 0.3 and 0.2 empirically based on the validation set. PReLU [30] with the negative slope of 0.2 is used as the activation function in (1).

**Training details** In the training phase, the Adam [31] is employed as the optimizer with a learning rate of 0.001. Batch size is set to 64 and training takes 600,000 iterations. Following [11], data augmentation methods mixup [32] and SpecAugment [33] are applied in our experiments to prevent the system from over-fitting and improve the performance.

### C. Experimental Results and Analysis

Table I demonstrates the performance of our proposed AT-GCN and other state-of-the-art methods on the Audioset. The results indicate that the proposed AT-GCN outperforms all the

[1]https://github.com/WangHelin1997/AT-GCN
[2]https://github.com/stanfordnlp/GloVe

TABLE II
ACCURACY COMPARISONS ON DIFFERENT CONSTRUCTION METHODS OF THE ADJACENCY MATRIX

| Construction Method | mAP | mAUC | d-prime |
|---|---|---|---|
| AT-GCN w/ method in [15], [28] | 0.431 | 0.973 | 2.727 |
| AT-GCN w/ method in [20] | 0.429 | 0.971 | 2.701 |
| AT-GCN w/o scheme | 0.132 | 0.907 | 1.872 |
| AT-GCN w/ scheme in (6) | 0.267 | 0.952 | 2.360 |
| AT-GCN w/ scheme in (7) | 0.188 | 0.932 | 2.113 |
| AT-GCN w/ scheme in (6) & (7) | **0.434** | **0.974** | **2.736** |

compared methods, which confirms the effectiveness of modeling the label dependencies. A single layer GCN learns the node representations by aggregating the information of the immediate neighbors guided by the graph structure information. The results show that AT-GCN with one GCN layer achieves a better performance than the baseline model (CNN10) owing to these statistical relations, which is similar to [16]. While in multi-layer GCN, information is propagated from more neighbors, which implicitly models the deep label dependencies and offers more performance gain. However, increasing the number of GCN layers may lead to over-smoothing [34], which may mix the features of too many nodes and make them indistinguishable. It can be observed that AT-GCN with two GCN layers provides a good trade-off in these aspects and achieves the best performance.

In order to analyze the impacts of the formulation of the adjacency matrix, ablation experiments are carried out. As shown in Table II, AT-GCN does not perform well when no re-weighting scheme is applied because of the over-fitting and over-smoothing problems, which are discussed in Section III-B. In [15], [28], a threshold is set to filter the noisy edges and other edges are treated equally, which is achieved by binarizing the adjacency matrix. However, our proposed re-weighting schemes retain the information of the other edges by only re-weighting the noisy edges and obtain better performance. Both the re-weighting schemes in (6) and (7) improve the performance, and a higher performance can be achieved with the combination of them. In addition, we have tested another construction method of the adjacency matrix [20], which utilizes the ontology rather than the statistical relations. Our proposed method achieves better performance, which shows that the label dependencies are more important in the large-scale multi-label dataset (*i.e.* Audioset).

In addition, we vary the values of the hyperparameters $\tau$ in (6) and $p$ in (7) to analyze the effects, and show the results in Fig. 3. As the values of $\tau$ and $p$ increase, the accuracy is boosted and then drops, which achieves the peek accuracy when $\tau = 0.3$ and $p = 0.2$. $\tau$ is the threshold to filter the edges, and the small values of $\tau$ mean the edges of small probabilities (*i.e.* noisy edges) are filtered. However, when too many edges are filtered, the correlated neighbors will be ignored as well which decreases the accuracy. $p$ determines the balance between a node itself and its neighbors when updating the node features. If $p$ is too small, the nodes of the graph cannot get sufficient information from correlated nodes. On the other hand, it will lead to over-smoothing if $p$ is too large.

Furthermore, the t-SNE [36] is adopted to visualize the last node representations (*i.e.* $\boldsymbol{H}^{(L)}$ in (3)) learned by our proposed AT-GCN as well as the weights of the last fully-connected layer of CNN10, with the results shown in Fig. 4. It can be observed that CNN10 can learn some meaningful relationships among the

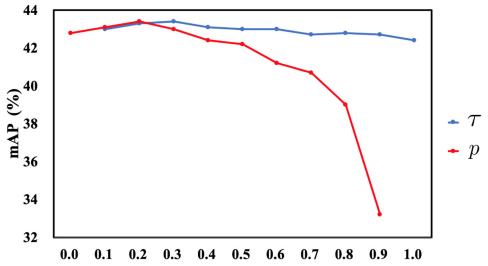
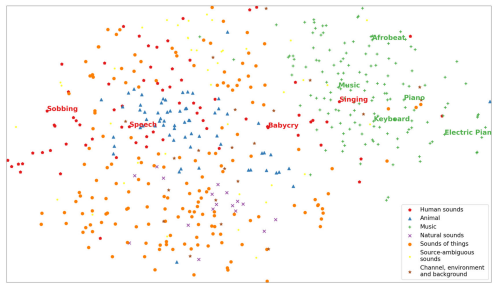
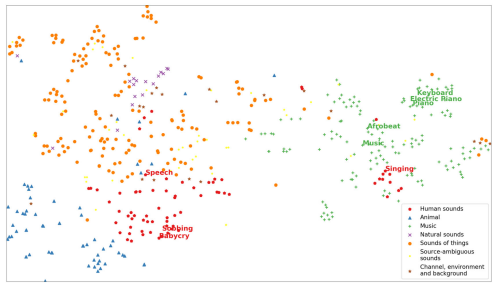



Fig. 3. Accuracy comparisons of different values of $\tau$ and $p$ for AT-GCN with two GCN layers (metric: mAP). Note that when $\tau = 1.0$, $\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}$ in (1) becomes the identity matrix. AT-GCN degenerates to a structure almost identical to CNN10, but with an extra fully-connected layer (as a GCN layer degenerates to a fully-connected layer). As a consequence, the mAP result (42.3%) by AT-GCN is slightly different from that of CNN10 (42.2%).




(a) t-SNE on the weights of the last fully-connected layer of CNN10




(b) t-SNE on the last node representations learned by AT-GCN

Fig. 4. Visualization of the last node representations learned by AT-GCN and the learned weights of the last fully-connected layer of CNN10 on Audioset. Note that Audioset [21] contains 527 classes (527 dots in the figure) with 7 general categories (the same color dots in the figure). Here, 9 classes are marked as examples.

labels, such as the cluster pattern of the general category *music*. However, the other cluster patterns are not clear and the semantic related labels (*e.g. sobbing* and *babycry*) are not close in the label space. In contrast, AT-GCN shows less divergence in the label space and exhibits more clear cluster patterns. Specifically, the semantic related labels (such as *piano* and *keyboard*, *sobbing* and *babycry*) tend to be much closer in the label space than CNN10, which shows the effectiveness of modeling the label dependencies.

## V. CONCLUSION

In this letter, a novel audio-tagging method (AT-GCN) has been proposed where the implicit dependencies between the labels are modeled by a graph convolutional network. Our proposed method achieves the state-of-art performance on the Audioset.

## REFERENCES


[1] T. Virtanen, M. D. Plumbley, and D. Ellis, *Computational Analysis of Sound Scenes and Events*. Berlin, Germany: Springer, 2018.

[2] R. Typke, F. Wiering, and R. C. Veltkamp, "A survey of music information retrieval systems," in *Proc. 6th Int. Conf. Music Inf. Retrieval*, 2005, pp. 153–160.

[3] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, "A survey of audio-based music classification and annotation," *IEEE Trans. Multimedia*, vol. 13, no. 2, pp. 303–319, Apr. 2010.

[4] H. Park and C. D. Yoo, "CNN-based learnable gammatone filterbank and equal-loudness normalization for environmental sound classification," *IEEE Signal Process. Lett.*, vol. 27, pp. 411–415, 2020.

[5] L. Ford, H. Tang, F. Grondin, and J. Glass, "A deep residual network for large-scale acoustic scene analysis," in *Proc. Interspeech*, 2019, pp. 2568–2572.

[6] K. Choi, G. Fazekas, and M. Sandler, "Automatic tagging using deep convolutional neural networks," in *Proc. 17th Int. Conf. Music Inf. Retrieval*, 2016, pp. 805–811.

[7] S. Hershey *et al.*, "CNN architectures for large-scale audio classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 131–135.

[8] E. Cakır, T. Heittola, and T. Virtanen, "Domestic audio tagging with convolutional neural networks," in *Proc. IEEE AASP Challenge Detection Classification Acoust. Scenes Events*, 2016.

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

[10] 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.

[11] 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," 2019, *arXiv:1912.10211*.

[12] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 2392–2396.

[13] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Large-scale weakly supervised audio classification using gated convolutional neural network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 121–125.

[14] 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.

[15] Y. Sun and S. Ghaffarzadegan, "An ontology-aware framework for audio event classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 321–325.

[16] K. Imoto and S. Kyochi, "Sound event detection using graph Laplacian regularization based on event co-occurrence," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 1–5.

[17] Y. Xu, Q. Huang, W. Wang, P. J. B. Jackson, and M. D. Plumbley, "Fully DNN-based multi-label regression for audio tagging," in *Proc. Workshop Detection Classification Acoust. Scenes Events*, Budapest, Hungary, Sep. 2016.

[18] M. Cartwright *et al.*, "SONYC Urban Sound Tagging (SONYC-UST): A multilabel dataset from an urban acoustic sensor network," in *Proc. Workshop Detection Classification Acoust. Scenes Events*, 2019, pp. 35–39.

[19] J. P. Bello *et al.*, "SONYC: A system for monitoring, analyzing, and mitigating urban noise pollution," *Commun. ACM*, vol. 62, no. 2, pp. 68–77, Feb. 2019.

[20] H. Shrivaslava, Y. Yin, R. R. Shah, and R. Zimmermann, "MT-GCN for multi-label audio tagging with noisy labels," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 136–140.

[21] 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.

[22] X. Li, F. Zhao, and Y. Guo, "Multi-label image classification with a probabilistic label enhancement model," in *Proc. Conf. Uncertainty Artif. Intell.*, 2014, vol. 1, pp. 430–439.

[23] Q. Li, M. Qiao, W. Bian, and D. Tao, "Conditional graphical lasso for multi-label image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2977–2986.

[24] C.-W. Lee, W. Fang, C.-K. Yeh, and Y.-C. Frank Wang, "Multi-label zero-shot learning with structured knowledge graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1576–1585.

[25] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *Proc. ICLR*, 2016.

[26] Z. Wu *et al.*, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 24, 2020, doi: 10.1109/TNNLS.2020.2978386.

[27] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3538–3545.

[28] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5177–5186.

[29] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1532–1543.

[30] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn.*, 2013, vol. 30, pp. 1–6.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2014.

[32] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proc. ICLR*, 2017.

[33] D. S. Park *et al.*, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019, pp. 2613–2617.

[34] G. Li, M. Muller, A. Thabet, and B. Ghanem, "DeepGCNs: Can GCNs Go as Deep as CNNs?" in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9267–9276.

[35] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.

[36] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.