Self-supervised Graphs for Audio Representation Learning with Limited Labeled Data

Amir Shirian, Krishna Somandepalli *Member, IEEE*, Tanaya Guha, *Member, IEEE,*

***Abstract*—Large-scale databases with high-quality manual labels are scarce in audio domain. We thus explore a self-supervised graph approach to learning audio representations from highly limited labelled data. Considering each audio sample as a graph node, we propose a subgraph-based framework with novel self-supervision tasks to learn effective audio representations. During training, subgraphs are constructed by sampling the entire pool of available training data to exploit the relationship between the labelled and unlabeled audio samples. During inference, we use random edges to alleviate the overhead of graph construction. We evaluate our model on three benchmark audio datasets spanning two tasks: acoustic event classification and speech emotion recognition. We show that our semi-supervised model performs better or on par with fully supervised models and outperforms several competitive existing models. Our model is compact and can produce generalized audio representations robust to different types of signal noise. Our code is available at `github.com/AmirSh15/SSL_graph_audio`**

***Index Terms*—Acoustic event classification, graph neural network, speech emotion recognition, self-supervised learning, semi-supervised learning, sub-graph construction.**

I. Introduction

Large databases with high-quality manual labels are scarce in audio domain. For tasks such as speech-based emotion analysis, manual labels are often difficult to acquire due to the subjectivity involved in the perception and expression of emotion across speakers, language and culture. On the other hand, for tasks such as acoustics event classification, manually labeling a large volume of audio data is simply impractical. Thus a core challenge in audio analysis is to learn from a limited amount of labeled data while taking advantage of larger amount of unlabeled training samples.

Why graphs? Self-Supervised Learning (SSL) has emerged as an effective approach to learning from unlabeled data [1]–[4]. We propose an SSL approach on graphs to learn effective audio representations from limited amount of labeled data. Considering each audio sample as a node in a graph, we cast audio classification as a node labeling task. The motivation behind adopting a graph approach is two-fold: (i) It leads to compact models as compared to commonly used recurrent speech models as noted in recent works [5], [6]; (ii) A graph structure, if properly constructed, can efficiently capture the relationship between the small number of available labeled nodes and a larger number of unlabeled nodes. Extensive experiments with standard benchmarks brings out the advantages of graph-based methods in terms of performance compared to the non-graph models.

Following the success of SSL on images, it has been extended to graph data for both fully supervised [7] and semi-supervised [8]–[10] tasks. Graph SSL tasks usually involve learning the local or global structure, or the context information in the data [7], [8], [10]. Conventional graph tasks such as node clustering and graph partitioning have already been used as SSL tasks [8]. In the audio domain, SSL has also started gaining popularity. Several recent papers report that SSL can improve over fully supervised models [11] while others use crossmodal self-supervision from visual domain [12], [13]. However, works that use graph approach to learning audio representation is limited. We are aware of only one recent work where an audio sequence has been considered as a line graph to exploit graph signal processing theory to achieve accurate spectral graph convolution [5].

In this paper, we propose a graph SSL approach to learning effective audio representations from limited amount of labelled data. Considering each audio sample as a graph node, we propose a subgraph-based learning framework with new self-supervision tasks. Our framework takes advantage of the entire pool of available data (labelled and unlabeled) during training; while during inference, our subgraphs are constructed using random edges with no overhead (e.g., nearest neighbour computation) of graph construction. In contrast to the more common SSL-then-finetune approach, we use an auxiliary learning paradigm where an SSL task and a node labelling task are performed jointly. Evaluation on large benchmark databases shows that our model achieves better results than fully supervised models outperforming state-of-the-art on several databases. To summarize, our contributions are as follows:

- We develop a subgraph-based auxiliary learning framework for audio representation learning with limited labelled data. To the best of our knowledge, ours is the first work on self-supervised semi-supervised audio representation learning with graphs.
- We propose a new graph SSL task, namely graph shuffling, and a new variant of graph denoising SSL task. We show that they can improve the performance of any graph network for semi-supervised node classification.
- We demonstrate the superior performance of our model for two tasks (acoustic event classification, and speech emotion classification) on three large benchmark audio databases. Our model, despite using limited labelled data, performs better or on par with fully supervised models, and can produce representations that are robust to various types of audio noise.

A. Shirian is with the Department of Computer Science, University of Warwick, UK. K. Somandepalli is with Google Research, US. T. Guha is with the School of Computing Science, University of Glasgow, UK.

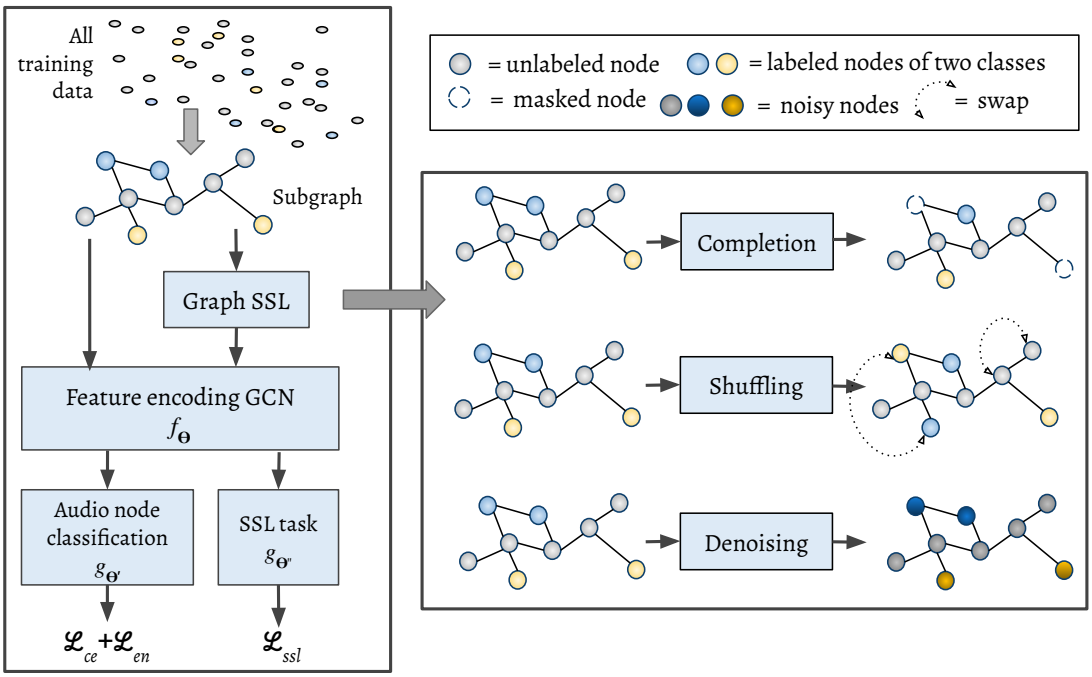


Fig. 1. Our model: A subgraph-based audio representation learning framework with SSL task. Our subgraph construction technique is efficient, can handle class-imbalance and the SSL framework facilitates robust and effective learning from highly limited labeled data.

II. Related Work

In this section, we review past works in the areas of self-supervised learning and graph neural networks (GNN) in the context of audio processing.

A. Self-supervised learning in audio

SSL has gained considerable popularity since its introduction in natural language processing [4] and computer vision [1]–[3] owing to its ability to learn effective data representations without requiring manual labels. Acquiring detailed manual labels is arguably more difficult (and often, expensive) in many audio and speech processing tasks, which makes SSL an increasingly popular paradigm in audio analysis. Contrast predictive coding (CPC) was the earliest work on SSL in the audio domain [14]. This work demonstrated the applicability of contrastive SSL to audio by predicting latent audio features at a future time instant. The popular model, Wav2Vec, further refined this CPC approach [15] to produce state-of-the-art audio presentations.

The majority of SSL works in audio domain rely on extracting audio descriptors and then utilising deep models to reconstruct a perturbed version of those descriptions as SSL tasks. For example, a self-supervised neural voice synthesizer was used [16] to reconstruct the input as an SSL (pretext) task. The Problem Agnostic Speech Encoder (PASE) [17] is another recent work that seeks to learn multitask speech representations from raw audio by predicting a number of handcrafted attributes like MFCCs and prosody features. Teacher-student models have also been investigated, where the trained model from the previous epoch serves as the teacher for the current epoch [18]. Several recent papers in audio analysis report that SSL can improve over fully supervised models [11], [19], [20] while others use crossmodal self-supervision from visual domain [12], [13].

B. Graph neural networks in audio

The predominant approach to audio/speech classification has been using Convolutional Neural Network (CNN) or Long-Short Memory Network (LSTM)-based models on a set of low-level descriptors. Works that use graphs to learn audio representation are limited, but steadily increasing. In a recent work, we have shown that graphs can be used to model audio samples effectively, leading to light-weight yet accurate models for emotion recognition in speech [21]. This work used a simple cycle and line graph to describe a given audio data sample. A follow-up work generalised such graph representation of audio to a learnable graph structure [22]. A graph-based neural network was utilised to capture the relationships within various speech segments of speakers in a conversation for speech emotion classification [23]. In another work, speech signals are represented as graphs to better capture the global feature representation for speech emotion recognition [24], where deep frame-level features are generated by an LSTM followed by a GNN to classify the graph representation of utterances. In another recent work [25], each audio channel is viewed as a node while constructing a speech graph for speech enhancement task. This allows for the discovery of spatial correlation among several channels. GNNs have been also employed in the context of fusing information from multiple heterogeneous modalities [26], [27].

In a recent study [28], an ontology-aware approach to acoustic event classification has been proposed that uses feedforward ontology layers and GCNs as two subnetworks. The intra-dependencies among labels are captured by the

feedforward ontology layers, while GCN layers focus on modelling the inter-dependency structure of labels. Another work [29] considers the likelihood of co-occurrences between acoustic events by first extracting audio features using a CNN-based network. Then, based on the frequency of audio node labels, a graph structure is created, with each node representing a label. The created graph is then used to train a GCN to learn node representations by propagating information between neighbouring nodes. Multitask GCN has also been used in literature [30] to alleviate the effect of label noise and utilise the hierarchical structure with successful results on audio tagging. Different from the previous studies, our current paper considers each audio sample as a graph node and presents a subgraph-based self-supervised, semi-supervised learning approach.

C. Graphs and SSL

SSL has been extended to graph data for both fully supervised [7] and semi-supervised [8]–[10] tasks. Graph SSL tasks usually involve learning the local or global structure, or the context information in the data [7], [8], [10]. Conventional graph tasks such as colouring and partitioning have already been used as SSL tasks [8]. However, we are not aware of any work that uses a graph approach to learning audio representation using SSL.

III. Proposed Approach

In this section, we propose our self-supervised (sub)graph-based audio representation learning model. Our model consists of an audio feature encoder, a subgraph construction step and a multitask-SSL architecture with new pretext tasks and loss functions.

A. Audio feature encoder

Our model has a feature encoder $f : \mathcal{S} \rightarrow \mathcal{Z}$ that takes raw audio $\mathcal{S}$ as input and returns embedding $\mathcal{Z}$. These embeddings are used as node attributes in graph $\mathcal{G}$ that we construct using labelled and unlabeled training data (described below). Owing to the different types of audio data (speech and sound), we use two different feature encoders: low-level descriptors for speech data and log-spectrogram based convolutional features for the generic (non-speech) audio. We chose simpler embeddings so as to demonstrate the effectiveness of our graph approach. Nevertheless, our model is not tied to any specific embedding, and rich audio embeddings such as *wav2vec* [15] may lead to better classification results. More details about f are provided in the Experiments section.

B. Graph construction during training

Given a collection of N (labelled and unlabeled) audio samples for training, we construct an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to capture the relationships among the samples, where $\mathcal{E}$ is the set of all edges between the connected nodes, and $\mathcal{V} = \mathcal{V}_l \cup \mathcal{V}_u$ has $|\mathcal{V}_l| = M$ labeled and $|\mathcal{V}_u| = (N - M)$ unlabeled nodes. To construct the graph $\mathcal{G}$, we first consider the labeled nodes. For a node $v_i \in \mathcal{V}_l$, we compute its k-nearest neighbors among *all* nodes in $\mathcal{V}_l$ based on their node attributes $\mathbf{z}_i$. We add an edge $e_{ij} \in \mathcal{E}$ with edge weight $a_{ij} = 1$ to the first Q nodes, if v_j is among the k-nearest neighbors of v_i and has the same label as v_i. We add another edge $e_{ip} \in \mathcal{E}$ with weight $a_{ip} = -1$ between v_i and its farthest node $v_p \in \mathcal{V}_l$. This negative weight is expected to force the two nodes to be apart in the embedding space. Every unlabeled node in $\mathcal{V}_u$ is connected to its two nearest and one farthest neighbors with respective edge weight of 1 and -1, where the neighbors can be any node in $\mathcal{V}$. We thus obtain a corresponding graph adjacency matrix $\mathbf{A} = \{a_{ij}\}_{i,j=1}^{N}$.

1) Subgraph construction:: Instead of constructing one large graph containing all training samples, we propose to construct and train on subgraphs. In our case, subgraph construction ensures that we do not end up with a *sparse graph*. This is also motivated by the observations made in several recent papers where subgraphs are able to learn local context more effectively (without oversmoothing or node dependence) while reducing computational load [31]–[34]. To construct a subgraph $\mathcal{G}_s$, we randomly select $N_s \in \mathcal{V}_l$ labeled nodes (equal number of samples from each class) and $M_s \in \mathcal{V}_u$ unlabeled training nodes, yielding a set of $\mathcal{V}_s$ nodes, $|\mathcal{V}_s| = N_s + M_s$. This procedure ensures the degrees of the nodes do not vary too much and class balance is maintained in each subgraph. Next, the edges in the subgraph are added the same way as for the full graph mentioned above. We also show that this approach produces better results than working with a single large graph.

C. Subgraph SSL training with limited labels

We adopt an auxiliary learning paradigm to merge self-supervision into the main task of audio classification. This is done by jointly optimizing for node classification and an auxiliary graph SSL task. Our model (see Fig. 1) has a shared GCN module for learning the latent audio representations, which is followed by two branches: one for audio classification and the other for SSL. Our model is *inductive*, i.e., neither attributes nor edges of the test nodes are present during the training process. This is a more challenging scenario than transductive learning [35].

Our node (audio) classification subnetwork uses supervision from only the true labels while producing *pseudolabels* for the unlabeled nodes. We train this sub-network using two loss functions:

(i) Cross-entropy loss $\mathcal{L}_{ce}$ computed for the *labeled* nodes.

$$\mathcal{L}_{ce} = -\sum_{v_p \in \mathcal{V}_l} \mathbf{y}_p \log(\hat{\mathbf{y}}_p) \tag{1}$$

(ii) For the *unlabeled* nodes, we propose to compute an **entropy regularization loss** $\mathcal{L}_{en}$. This can be considered as a measure of class overlap - the lower the entropy loss, the more distinguishable the predicted class labels are.

$$\mathcal{L}_{en} = -\sum_{v_p \in \mathcal{V}_u} P(\hat{\mathbf{y}}_p) \log(P(\hat{\mathbf{y}}_p)) \tag{2}$$

where, $\mathbf{y}_p$ is the true label for node v_p, $\hat{\mathbf{y}}_p$ is its predicted label, $\mathcal{V}_l$ and $\mathcal{V}_u$ are the sets of labeled and unlabeled training

Algorithm 1: Subgraph-based SSL training

Input : Labeled nodes $\mathcal{V}_l$, unlabeled nodes $\mathcal{V}_u$, node embeddings $\mathbf{Z}$

Output: Learned parameters $(\Theta, \Theta', \Theta'')$, pseudolabels $\hat{\mathbf{y}}_p$ for $\mathcal{V}_u$

for *each epoch* **do**
- $\mathcal{G}_s \leftarrow \texttt{subgraphConstruct}(\mathcal{V}_l, \mathcal{V}_u, \mathbf{Z})$
- $\hat{\mathcal{G}}_s \leftarrow \texttt{createGraphforSSL}(\mathcal{G}_s)$
- $\hat{\mathbf{y}} \leftarrow g_{\Theta'}(f_{\Theta}(\mathcal{G}_s)); \quad \tilde{\mathbf{Z}} \leftarrow g_{\Theta''}(f_{\Theta}(\hat{\mathcal{G}}_s))$
- $\Theta, \Theta', \Theta'' \leftarrow \mathcal{L}_{ce} + \lambda_1 \mathcal{L}_{en} + \lambda_2 \mathcal{L}_{ssl}$

end

Function `subgraphConstruct` $(\mathcal{V}_l, \mathcal{V}_u, \mathbf{Z})$
- $\mathcal{V}_s \leftarrow$ randomly select N_s nodes from $\mathcal{V}_l$ and M_s nodes from $\mathcal{V}_u$
- **for** $\forall v_s \in \mathcal{V}_s$ **do**
  - $\mathcal{V}_{nn} \leftarrow$ nearest neighbors(v_s)
  - **if** $v_s \in \mathcal{V}_l$ **then**
    - $\mathcal{E}_s \overset{+}{\leftarrow}$ edge between v_s and 2 nearest nodes $v_i \in \mathcal{V}_{nn}$ with same labels with edge weight 1
  - **else**
  - **end**
  - $\mathcal{E}_s \overset{+}{\leftarrow}$ edge between v_s and 2 nearest nodes $v_i \in \mathcal{V}_{nn}$ with edge weight 1
- **end**
- $\mathcal{E}_s \overset{+}{\leftarrow}$ edge between v_s and farthest $v_i \in \mathcal{V}_{nn}$ with weight (-1)

end

return $\mathcal{G}_s(\mathcal{V}_s, \mathcal{E}_s)$

samples. The SSL subnetwork is trained using a graph SSL task optimizing over a loss function $\mathcal{L}_{ssl}$ (discussed below). Given a subgraph input $\mathcal{G}_s$, the overall optimization is given by:

$$\min_{\Theta,\Theta',\Theta''} \big[\mathcal{L}_{ce}(\Theta, \Theta', \mathcal{G}_s) + \lambda_1 \mathcal{L}_{en}(\Theta, \Theta', \mathcal{G}_s) + \lambda_2 \mathcal{L}_{ssl}(\Theta, \Theta'', \hat{\mathcal{G}}_s)\big] \tag{3}$$

where Θ, Θ', and Θ'' are the learnable parameters for the shared GCN, classification GCN and the SSL sub-networks, $\hat{\mathcal{G}}_s$ is the SSL variant of $\mathcal{G}_s$, and λ_1 and λ_2 control the relative weights of SSL loss and entropy regularization.

To this end, we experiment with three graph SSL tasks. These SSL tasks are *model-agnostic* and can be used with any graph neural network. We propose a novel proxy task in the graph SSL domain: graph shuffling. In addition, we also experiment with the recently introduced graph completion and graph denoising proxy task.

1) Graph denoising: Motivated by past works [36], [37], we employ a new variant of the SSL task of graph denoising. Given a subgraph $\mathcal{G}_s$, we construct a noisy graph $\hat{\mathcal{G}}_s$, where Gaussian noise is added to every node feature vector $\hat{\mathbf{z}}_s(i) = \mathbf{z}_s(i) + \mathbf{x}(i)$, where $\mathbf{x}(i) \sim \mathcal{N}(0, \epsilon)$ by adding a Gaussian noise with zero mean and ϵ variance to each node feature. The SSL regression task is to learn to reconstruct node feature matrix $\mathbf{Z}_s$ from noisy $\hat{\mathbf{Z}}_s$ by optimizing the following loss function:

$$\mathcal{L}_{ssl} = \frac{1}{|\mathcal{V}_s|} \|\tilde{\mathbf{Z}}_s - \mathbf{Z}_s\|_F^2 \text{ where } \tilde{\mathbf{Z}}_s = g(\Theta, \Theta'', \hat{\mathbf{Z}}_s) \tag{4}$$

It is worth mentioning that in a recent study [37], a denoising autoencoder has been used to perform SSL on graph data. This work is different from ours; they used a self-supervised approach with cross-entropy loss that jointly learns the graph structure and node features at the same time. while we apply the cleaning loss (Eq. 4) in a fixed graph after a graph-based feature extractor to reconstruct the node features.

2) Graph completion:: This SSL task forces the network to learn to reconstruct missing information so as to learn local context. Following a recent work [8], we mask a random set of target nodes $\mathcal{V}_{sc} \subset \mathcal{V}_s$ by setting their node attributes to zero. The task is to recover this missing information for the target nodes. Given $\mathcal{G}_s$, denote the ground truth feature matrix corresponding to $\mathcal{V}_{sc}$ as $\mathbf{Z}_{sc}$ and its predicted version as $\tilde{\mathbf{Z}}_{sc}$, the SSL loss is then given by

$$\mathcal{L}_{ssl} = \frac{1}{|\mathcal{V}_{sc}|} \|\tilde{\mathbf{Z}}_{sc} - \mathbf{Z}_{sc}\|_F^2 \tag{5}$$

3) Graph shuffling (proposed):: We propose a novel SSL task that aims to determine whether or not a graph node is in its correct position. This task encourages the graph network to learn structural dependencies among nodes without using the available labels. Given $\mathcal{G}_s$, we randomly sample a set of graph nodes $\mathcal{V}_{sh} \subset \mathcal{V}_s$ and shuffle their node attributes randomly with each other creating $\hat{\mathcal{G}}_s$. The SSL task is posed a binary node classification task on $\hat{\mathcal{G}}_s$ where the model outputs 1 if is node is unchanged and 0 otherwise.

$$\mathcal{L}_{ssl} = -\frac{1}{|\mathcal{V}_{sh}|} \sum_{v_i \in \mathcal{V}_{sh}} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \tag{6}$$

D. Subgraph construction during inference

After completing training, we obtain a large number of *pseudolabeled* samples. In order to create the subgraph during inference, we randomly sample $|\mathcal{V}_s|$ nodes (equal number of nodes from each class) from the entire set of training samples considering both true and psesuolabels. The edges between these nodes are constructed as was done during training. Each test node (audio sample) v_t has to be connected to this graph structure $\mathcal{G}_s$. Computing nearest neighbours during inference time may not always be practical; hence, we propose to connect v_t with T training nodes (labelled or pseudolabeled) *randomly* with edge weight 1.

Please note that during interference time, we do not need to compute nearest neighbors of the test nodes as the test nodes are connected randomly to the nodes from the training set. For the nodes in the inference subgraph coming from the training set, the nearest neighbours only come from the training set - information we already have precomputed and stored from the beginning. In summary, no new nearest neighbour computing is needed during inference. This is by design, precisely to avoid storing embeddings/labels and computing nearest neighbours again. Therefore we only need to store the pairwise distances of the training nodes in the RAM, not embeddings

or labels. This makes our method lightweight, which may be suitable even for edge devices.

1) Optimal number of random edges: A natural question is what is the optimal value of T so as to ensure that we do not connect to only pseudolabeled nodes, which could be incorrect. Hence, we ask: *What is the minimum number of nodes a test node, v_t, should be connected with, such that there is at least one connection with a true-labelled node?*

Let $\mathcal{G}$ be a graph with $|\mathcal{V}|$ nodes including known N true labelled and M pseudolabeled nodes. The probability of having all T edges from v_t to be connected with only the pseudolabeled nodes is given by $\frac{\binom{M-1}{T}}{\binom{N+M-1}{T}}$ using Hypergeometric distribution. Therefore, the probability of an v_t to be connected to at least one true labeled node is given by $P = 1 - \frac{\binom{M-1}{T}}{\binom{N+M-1}{T}}$. With known N and M, we set a high value of $P(= 0.9)$ to compute the value of T for our experiments. Once the graph is thus constructed, we use only the audio classification branch to determine the class labels for the test nodes.

IV. Experiments

This section presents extensive experimental results on the analysis of our model and demonstrates its effectiveness for audio classification tasks.

A. Semi-supervised acoustic event classification

1) Datasets: We use a large scale weakly labeled database called the **AudioSet** [38] that contains audio segments from YouTube videos. We work with 33 class labels that have a high rater confidence score (≥ 0.7) (see Fig. 2 for the names of those classes). This yields a training set of around 89,000 audio clips and a test set of more than 8,000 audio clips. We consider only 10% of the 89,000 training samples as labelled and the rest are used as unlabeled training data for our experiments.

2) Feature encoder: To extract the node features, each audio clip is divided into non-overlapping 960 ms segments. For each segment, a log-mel spectrogram is computed by taking its short-time Fourier transform using a frame of length 25 ms with 10 ms overlap, 64 mel-spaced frequency bins and log-transforming the magnitude of each bin. This creates log-mel spectrograms of dimension 96×64 which are the input to the pre-trained VGGish network [39]. We use the 128-dimensional features extracted from the VGGish for each log-mel spectrogram and average over all segments to form the final vector representation of each audio clip. Note that although we use VGGish embeddings to be comparable with previous works, other generic audio embeddings will work as well.

3) Experimental settings: To construct the subgraph during training, we sample data with $M_s = 5$, $N_s = 2 \times$ number of classes, and $T = 4$. It is worth noting that the T value is calculated using the probabilistic equation in section III.D, with $P > 0.99$. We also tested this hypothesis experimentally. From $T = 1$ to 20, we steadily increase the value of T. Each step improves classification performance by a significant margin until $T = 4$, where we get 63.8% accuracy. Increasing the T value further does not result in any significant

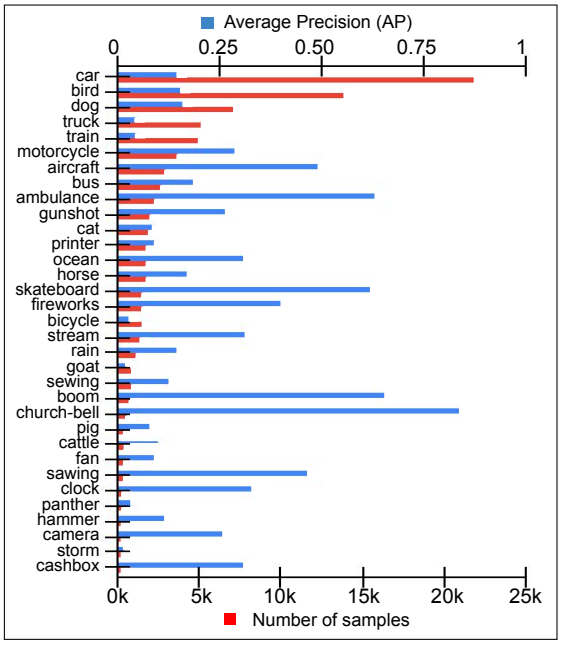


Fig. 2. Results on **AudioSet**: Class distribution and average precision per class achieved using our model with graph completion task. Note that our model can achieve high AP even for classes with fewer samples owing to our graph construction process.

performance improvement. This yields a subgraph size of 71 nodes. This process ensures the subgraph nodes are class-balanced every time. The subgraph construction process is repeated until all unlabeled nodes appear at least once i.e., the subgraphs are constructed by sampling without replacement. Note that for constructing the edges, the nearest neighbours need to be computed only once for all the training nodes. For the test nodes, no nearest neighbour computation is needed due to our random edge construction strategy. For graph neural network, we select regular GCN [40] with 2 graph convolution layers and a hidden size of 256 for all layers. We used the same architecture for all experiments. We use the 80:10:10 train:validation:test split *only* for the semi-supervised framework, where we consider 10% of that 80% train data as labelled and the rest as unlabeled. All hyperparameters were chosen solely based on validation data, with test data accounting for no model or parameter selections. In order to obtain robust estimates of performance metrics, this process is repeated 5 times to report the average accuracy and standard deviation. We set $\lambda_1 = 0.01$ and $\lambda_2 = 0.1$ (see Eq. 3). Our model uses Xavier initialization and the Adam optimizer with a learning rate of 0.001 for all experiments. We use Pytorch for implementing our models and the baselines. All models were trained on a single NVIDIA RTX-2080Ti GPU.

4) Baselines: We compare our method with a number of fully supervised models in Table I. The Spectrogram-VGG model is the same as configuration A in [41] with only one change: the final layer being a softmax with 33 units. The feature for each audio input to the VGG model is a log-mel spectrogram of dimension 96×64 computed averaging across non-overlapping segments of length 960ms. We did not adjust the hyperparameters for baselines other than VGG and used the settings stated in the original papers. The fully supervised

TABLE I
ACOUSTIC EVENT CLASSIFICATION RESULTS ON **AUDIOSET**: OUR MODEL, USING ONLY 10% LABELED DATA, OUTPERFORMS ALL SEMI-SUPERVISED MODELS AND SEVERAL FULLY SUPERVISED MODELS. THE FULLY SUPERVISED VERSION OF OUR MODEL WITHOUT SSL SHOWS COMPARABLE PERFORMANCE TO THE STATE OF THE ART, INDICATING THE EFFECTIVENESS OUR SUBGRAPH-BASED LEARNING STRATEGY FOR AUDIO CLASSIFICATION.PLEASE NOTE THAT THE PARAMETERS FOR OUR MODEL EXCLUDES THE FEATURE EXTRACTOR, WHICH VARIES DEPENDING ON THE FEATURES USED.

Model	**mAP**	**Params**
Semi-supervised		
Ours w/o SSL	0.23 ± 0.01	218K
Ours w/ denoise	0.26 ± 0.00	260K
Ours w/ completion	0.27 ± 0.01	260K
Ours w/ shuffle	0.24 ± 0.00	219K
Ours w/ all three SSL	$\mathbf{0.28} \pm 0.02$	261K
Spectrogram-VGG	0.16 ± 0.05	6M
AST [42]	0.22 ± 0.01	88M
Fully supervised		
Ours w/o SSL	0.42 ± 0.02	218K
Spectrogram-VGG	0.26 ± 0.01	6M
DaiNet [43]	0.25 ± 0.07	1.8M
Wave-Logmel [44]	0.43 ± 0.04	81M
VATT [45]	0.39 ± 0.02	87M
AST [42]	$\mathbf{0.44} \pm 0.00$	88M

version of our model follows the same graph construction strategy as proposed with 80:10:10 (train:validation:test) split. The split process is done 5 times for our models and average performance with standard deviation is reported. All other baseline implementations are done by the authors and follow the same evaluation protocol as the proposed methods.

5) Results: Table I reports the average recognition accuracy (averaged over 5 runs) with standard deviation for each model and their variants. For the case of all three SSL, we applied all three self-supervised tasks in parallel with a shared feature encoder and add the corresponding loss functions to the main classification loss. It compares the performance of our model with different SSL tasks with that of fully supervised models in terms of mean Average Precision (mAP). The graph models with SSL outperform the plain graph model without SSL. When compared with the fully supervised models, our graph SSL models (denoise and completion in particular) outperform Spectrogram-VGG and DaliNet [43]. Our model also has significantly fewer learnable parameters. The fully supervised GCN model uses the same graph construction method as proposed and performs very close to the state-of-the-art. This demonstrates the effectiveness of our graph construction strategy. Fig. 2 shows the average precision (AP) per class for the AudioSet database. A high AP is achieved even for classes with fewer samples, e.g., *church-bell* has an AP of 0.843 even with only 627 samples. This suggests that our model is not highly affected by a lower number of samples owing to how we construct the sub-graphs during the training process.

TABLE II
SPEECH EMOTION RECOGNITION RESULTS: CLASSIFICATION (UNWEIGHTED) ACCURACY (IN %) ON TWO BENCHMARK DATABASES ARE PRESENTED. OUR MODEL, USING ONLY 10% LABELED DATA, OUTPERFORMS SEMI-SUPERVISED AND SEVERAL FULLY SUPERVISED MODELS ON BOTH DATABASES. THE FULLY SUPERVISED VERSION OF OUR MODEL PRODUCES THE HIGHEST ACCURACY EVEN WITHOUT SSL, INDICATING THE EFFECTIVENESS OF OUR SUBGRAPH-BASED LEARNING STRATEGY. (* INDICATES AUDIOVISUAL MODELS).

Model	IEMOCAP	MSP-IMPROV	**Param**
Semi-supervised			
Ours w/o SSL	63.8 ± 2.2	58.6 ± 1.8	212K
Ours w/ denoise	68.0 ± 1.1	64.1 ± 1.0	271K
Ours w/ completion	66.4 ± 1.7	63.8 ± 1.5	271K
Ours w/ shuffle	65.9 ± 1.4	64.1 ± 1.3	213K
Ours w/ all three SSL	$\mathbf{68.6} \pm 1.2$	$\mathbf{65.2} \pm 1.8$	272K
LadderNet [46]	60.7	-	-
Transformer* [47]	61.2	-	-
SimCLR [48]	65.1	-	30M
Self-supervised (non-graph)			
SSAST [49]	59.6	-	89M
BYOL-S/CvT [50]	65.1	-	5M
Wav2vec2.0 [51]	65.6	-	317M
HuBERT [51]	67.6	-	316M
Fully supervised			
Ours w/o SSL	**70.5**	**66.7**	212K
SegCNN [52]	64.5	-	-
GA-GRU [53]	63.8	55.4	-
CNNattn [54]	66.7	-	-
WADAN [55]	64.5	-	-
SpeechGCN [5]	62.3	57.8	30K

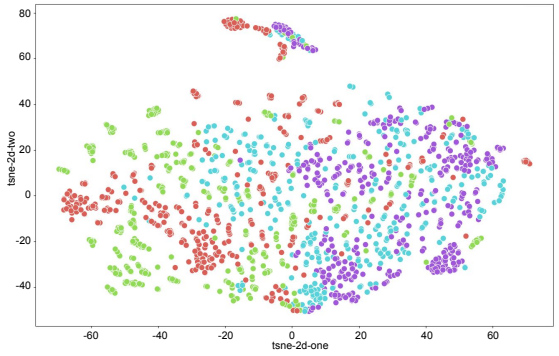
B. Semi-supervised speech emotion recognition

1) Datasets: For this task, we use the two most popular speech emotion datasets.

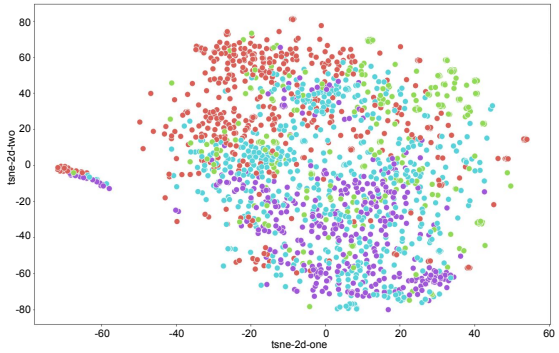
The **IEMOCAP** [56] dataset contains 12 hours of speech collected over 5 dyadic sessions with 10 subjects. It includes 4490 utterances with labels, 1103 *anger*, 595 *joy*, 1708 *neutral* and 1084 *sad*.

The **MSP-IMPROV** [57] contains 7798 utterances from 12 speakers collected across six sessions including 792 samples for *anger*, 3477 for *neutral*, 885 for *sad* and 2644 samples for *joy*. The train-test split is the same.

2) Feature encoder: Following relevant past work on speech emotion analysis [58], we extract a set of low-level descriptors (LLDs) from the speech utterances using the OpenSMILE library [59]. This feature set includes Mel-Frequency Cepstral Coefficients (MFCCs), zero-crossing rate, voice probability, fundamental frequency (F0) and frame energy. For each audio sample, we use a sliding window of length 25ms (with a stride length of 10ms) to extract the LLDs. The local features are smoothed temporally using a moving average filter, and the smoothed version is used to compute their respective first-order delta coefficients. Then we compute the mean, max, standard deviation, skew and kurtosis of the extracted LLDs and their delta coefficients to compute one feature vector per speech sample. Altogether this yields node



(a)



(b)

Fig. 3. Comparison of t-SNE plots and the Silhouette score for (a) our model (Silhouette score = 0.147) and (b) non-graph model SSAST (Silouette score - 0.013). The Siloutte scores being a measure of clustering quality indicates that our model learns more discriminative embeddings.

embeddings of dimension 165 for an audio clip.

3) Experimental settings: We follow the same settings as in the acoustic event classification task with a change in the graph size. The input graph used for this task has 53 nodes, where $M_s = 5$, $N_s = (12 \times \text{number of classes})$, and $T = 4$. Further discussion on the effect of graph size is presented in the next section.

4) Results: Table II compares our model against the state-of-the-art supervised, (non-graph) semi-supervised, and (non-graph) self-supervised methods on the two speech databases. Clearly, our method (with denoise SSL in particular) outperforms others by a significant margin, using only 10% of the training data as labelled. Again note that the fully-supervised version of the model yields comparable results with the state-of-the-art. Furthermore, as compared to non-graph self-supervised methods, our method provides comparable or higher results (especially with denoise and all three SSL) with a substantially reduced number of parameters as shown in Table II. We also showed graph-based and non-graph (SSAST) features [49] using the t-SNE plot . For comparison, we used the Silhouette score, which is a tool for evaluating clustering quality. As seen in Fig. 3, our method produces significantly better clusters with a 0.147 Silhouette score when compared to the non-graph method with a 0.013 Silhouette score. The results show that SSL with subgraph is an effective learning framework for speech classification even when labelled data is highly limited.

TABLE III
TO SHOW THE EFFECTIVENESS OF OUR PROPOSED FRAMEWORK WHEN THE GRAPH STRUCTURES ARE *known*, WE PRESENT SEMI-SUPERVISED NODE CLASSIFICATION RESULTS (IN % ACCURACY) ON BENCHMARK GRAPH DATABASES. THIS ESSENTIALLY DISENTANGLES THE EFFECT OF OUR PROPOSED GRAPH CONSTRUCTION METHODOLOGY FROM THE SSL-BASED SEMI SUPERVISED MODEL. THE RESULTS SHOW THAT SSL TASKS IMPROVE OVER BASIC MODELS IN ALMOST ALL CASES IRRESPECTIVE OF THE GRAPH NETWORK USED.

	SSL task	Cora	Citeseer	Pubmed
GCN	✗	80.9 ± 0.6	70.7 ± 0.6	79.1 ± 0.5
	denoise	81.1 ± 0.8	71.1 ± 0.8	78.4 ± 0.8
	completion	$\mathbf{81.6} \pm 0.8$	$\mathbf{71.6} \pm 0.6$	$\mathbf{79.2} \pm 0.7$
	shuffle	$\mathbf{81.6} \pm 0.6$	70.1 ± 1.1	78.4 ± 0.6
GAT	✗	83.1 ± 0.5	72.1 ± 0.6	77.5 ± 0.4
	denoise	$\mathbf{84.2} \pm 1.0$	$\mathbf{73.1} \pm 0.5$	$\mathbf{78.2} \pm 0.5$
	completion	84.2 ± 0.4	72.8 ± 1.1	78.0 ± 0.5
	shuffle	83.2 ± 0.9	72.6 ± 0.7	77.7 ± 0.4
GIN	✗	77.2 ± 0.5	68.1 ± 0.7	77.0 ± 0.4
	denoise	$\mathbf{78.9} \pm 0.8$	69.1 ± 1.4	77.2 ± 0.3
	completion	78.8 ± 0.6	$\mathbf{69.8} \pm 1.2$	$\mathbf{77.7} \pm 0.3$
	shuffle	78.6 ± 1.1	69.3 ± 0.8	77.3 ± 0.4
GraphMix	✗	83.8 ± 0.8	74.3 ± 0.7	80.6 ± 0.6
	denoise	$\mathbf{84.4} \pm 0.7$	$\mathbf{75.2} \pm 0.5$	81.4 ± 0.4
	completion	84.4 ± 0.7	74.3 ± 0.7	81.2 ± 0.3
	shuffle	84.2 ± 0.4	74.4 ± 0.6	$\mathbf{81.9} \pm 0.3$

C. Model analysis and ablation studies

1) Ablating graph construction: We next attempt to understand the effectiveness of self-supervision on transductive semi-supervised node classification. This requires experimenting with databases where the graph structures are known so as to disentangle the effect of our proposed graph construction methodology. We use three graph citation databases **Cora** (2708 nodes, 7 classes) and **Citeseer** (3327 nodes, 6 classes) [60] and **Pubmed** (19717 nodes, 3 classes) [61] following the standard benchmarking framework [62].

To verify the universality of our SSL tasks, we conduct experiments on several state-of-the-art graph neural networks: (i) Standard GCN [40], (ii) Graph Attention Network (GAT) [63] - a powerful variant of GCN, (iii) Graph Isomorphism Network (GIN) [64] and (iv) GraphMix [65]. Based on the graph models, we use a joint learning framework where the SSL task as an auxiliary task and node classification is the primary task. Table III compares the performance of the above graph models when augmented with the three SSL tasks. Clearly, the SSL tasks (particularly, denoising and completion) improve the accuracy of all models across all databases cases. As mentioned before, the SSL tasks we consider are model-agnostic and thus can be used to enhance any graph model. Also, note that these experiments use 16-dimensional embedding while the emotion recognition task used 165-dimensional embeddings and acoustic even classification used 1024-dimensional embeddings. This shows that our model works well with different types and dimensions of embeddings.

2) Why subgraph instead of a large graph: To investigate that subgraphs indeed produce superior performance,

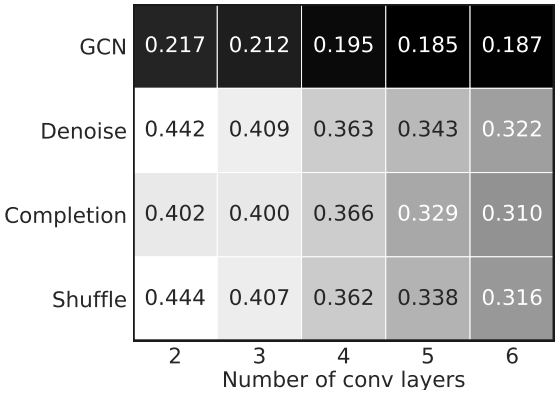


Fig. 4. Oversmoothness analysis for IEMOCAP dataset: Measured in terms of mean average distance (MAD), is significantly smaller for our SSL-based models compared to GCN models. Darker color indicates smaller MAD values i.e., higher oversmoothing. We observed a similar trend for other datasets studied in our work.

TABLE IV
MODEL ROBUSTNESS AGAINST NOISE: PERFORMANCE CHANGES IN RECOGNITION ACCURACY (IN %) FOR NOISY SPEECH VS. CLEAN SPEECH. THE RESULTS SHOWN ARE FOR THE IEMOCAP DATABASE WHERE ↓ INDICATES A DROP IN PERFORMANCE.

	Speech noise types				
	Babble	Factory	White	Pink	Cockpit
Ours w/o SSL	2.8 ↓	1.6 ↓	3.1 ↓	2.0 ↓	1.1 ↓
Ours w/ denoise	**0.5** ↓	1.2 ↓	**0.5** ↓	**0.9** ↓	**0.8** ↓
Ours w/ completion	0.7 ↓	1.1 ↓	0.8 ↓	1.3 ↓	1.6 ↓
Ours w/ shuffle	0.8 ↓	**0.8** ↓	1.0 ↓	1.5 ↓	0.9 ↓

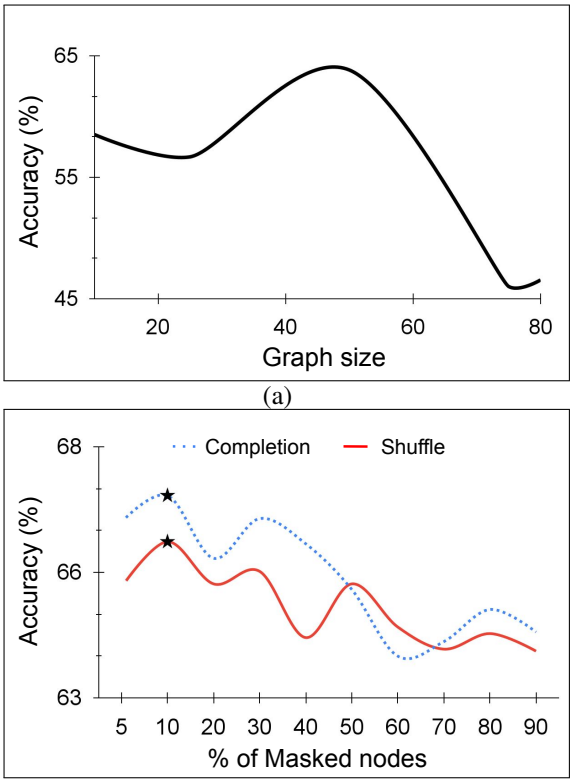


Fig. 5. (a) Impact of graph size on audio classification performance on the IEMOCAP database. (b) Effect of the fraction of masked nodes for the SSL tasks. Results shown are for IEMOCAP database. The ⋆ indicates best performance.

we experimented with a single big graph that includes all training samples simultaneously. When compared with semi-supervised audio node classification task on IEMOCAP, the large graph achieves only a 49.44% recognition accuracy which is much lower than the accuracy (63.84%) achieved using the subgraphs-based learning framework, both without SSL. An intuitive explanation of the superior performance of subgraphs for audio classification: Since the graph structure is not given, constructing smaller graphs with balanced class samples introduces less error (as compared to a large graph) as subgraphs limit the number of unlabeled nodes per graph. Fig. 5(a) shows how the classification performance varies with graph size (number of nodes). We observe that initially the recognition accuracy improves as the graph size increases (up to size 50), but then starts decreasing. Overall, our observation is that subgraphs are more effective than using larger graphs in settings where we have limited labeled and a large amount of unlabeled training data.

3) Number of masked nodes in SSL tasks: For the completion and the shuffle SSL tasks, we select a fraction of nodes (masked nodes) within a subgraph to apply the transformation. Fig. 5(b) shows how the fraction of masked nodes affects the classification performance. We observe that in general, the recognition accuracy drops as masked nodes are increased beyond 10%.

4) Oversmoothness: GCN models are known to suffer from oversmoothness as the number of layers increases [66]. We investigate whether our model benefits from SSL in mitigating oversmoothness. A quantitative metric for measuring oversmoothness is Mean Average Distance (MAD). It measures the average distance from a node to all other nodes [66]. Using MAD, we experimented with a varying number of graph convolution layers on the IEMOCAP database. Fig. 4 shows that SSL-based models are more resilient to oversmoothness, producing more distinguishable node embeddings. This is clear from the average distances being much larger compared to the no SSL case. A similar trend is observed in other datasets.

5) Robustness against noise: To investigate the robustness of our model against noisy data, we experiment with six common noise types that may corrupt speech data: babble, factory, white, pink and cockpit noise [67]. Assuming additive noise, the noisy mixtures are obtained by adding the noise (one type at a time) to each speech sample first and then using the feature encoders as usual. Noise is added only to the test samples during inference. We compare the performance of our model on noisy test data with that with clean test input in Table IV for the IEMOCAP dataset. Clearly, SSL provides significant robustness against noise as their drop in performance is consistently smaller compared to GCN semi-supervised results without SSL.

6) Reducing labelled data further: We further investigate how the subgraph strategy holds up to even more scarce labelled data, and whether or not the SSL tasks help. Fig. 6 shows the performance of our model with 2% and 5% labeled

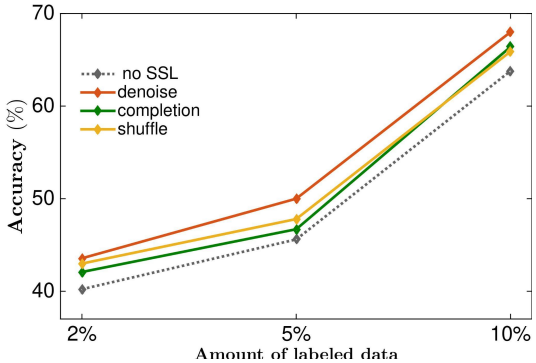


Fig. 6. Performance using less than 10% labeled data: Our model performs reliably with SSL producing consistent improvement as labeled data becomes scarce; the denoise task performs the best (results shown for IEMOCAP).

data in addition to 10% on the IEMOCAP database. We note that SSL tasks bring consistent improvement across all cases, where the denoise task performs the best.

V. Conclusion

Our work contributes to the understanding of semi-supervised audio representation learning - a relatively understudied topic in the acoustics and speech community. We developed a subgraph-based SSL framework for audio representation learning with limited labelled data. We make use of graphs to capture the underlying information in the unlabeled training samples and their relationship with the labelled samples. To this end, we proposed an effective subgraph construction technique and a new graph SSL task (graph shuffling). Our framework is generic, and can effectively handle both speech and non-speech audio data. Our model could achieve comparable or better performance than fully supervised models, despite using only 10% of the labelled data. Since the graph structure in our task has to be constructed first, our model is currently not end-to-end learnable. This could be addressed in future work where the graph structure itself is learned jointly with the embeddings. Our current model relies on pre-trained embeddings, which gives the flexibility of choosing any suitable embeddings given a task. Nevertheless, our model can be made end-to-end trainable which will be addressed in a future work.

Ethical/social impact statement: Our work may be applied to classify speech and acoustics data. Therefore, it may be used to 'hear' and should be used carefully. We used public databases that are mostly balanced in terms of male and female subjects. However, they are not balanced considering factors such as ethnicity, language spoken and other demographic factors. The speech emotion analysis application uses only four archetypal expressions. We are aware that human emotion is far more complex, and thus do not advocate the use of such systems for sensitive decision making areas. We also note that automatic emotion classification from speech is an evolving area of research and questions of fairness and universality remain to be explored.

References

[1] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, pages 1422–1430, 2015.

[2] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E. Hinton. Big self-supervised models are strong semi-supervised learners. In *NeurIPS*, 2020.

[3] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10687–10698, 2020.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Association for Computational Linguistics (NAACL)*, pages 4171–4186, 2019.

[5] Amir Shirian and Tanaya Guha. Compact graph architecture for speech emotion recognition. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6284–6288, 2021.

[6] Jiawang Liu and Haoxiang Wang. Graph isomorphism network for speech emotion recognition. In *Annual Conference of the International Speech Communication (Interspeech)*, pages 3405–3409, 2021.

[7] Anton Tsitsulin, Davide Mottin, Panagiotis Karras, Alex Bronstein, and Emmanuel Müller. SGR: Self-supervised spectral graph representation learning. *arXiv preprint arXiv:1811.06237*, 2018.

[8] Yuning You, Tianlong Chen, Zhangyang Wang, and Yang Shen. When does self-supervision help graph convolutional networks? In *International Conference on Machine Learning (ICML)*, pages 10871–10880, 2020.

[9] Qikui Zhu, Bo Du, and Pingkun Yan. Self-supervised training of graph convolutional networks. *arXiv preprint arXiv:2006.02380*, 2020.

[10] Wei Jin, Tyler Derr, Haochen Liu, Yiqi Wang, Suhang Wang, Zitao Liu, and Jiliang Tang. Self-supervised learning on graphs: Deep insights and new direction. *arXiv preprint arXiv:2006.10141*, 2020.

[11] Marco Tagliasacchi, Beat Gfeller, Félix de Chaumont Quitry, and Dominik Roblek. Pre-training audio representations with self-supervision. *IEEE Signal Processing Letters*, 27:600–604, 2020.

[12] Abhinav Shukla, Stavros Petridis, and Maja Pantic. Does visual self-supervision improve learning of speech representations for emotion recognition. *IEEE Transactions on Affective Computing*, 2021.

[13] Arsha Nagrani, Joon Son Chung, Samuel Albanie, and Andrew Zisserman. Disentangled speech embeddings using cross-modal self-supervision. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6829–6833, 2020.

[14] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[15] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. In *Annual Conference of the International Speech Communication (Interspeech)*, pages 3465–3469, 2019.

[16] Hyeong-Seok Choi, Juheon Lee, Wansoo Kim, Jie Lee, Hoon Heo, and Kyogu Lee. Neural analysis and synthesis: Reconstructing speech from self-supervised representations. In *NeurIPS*, 2021.

[17] Santiago Pascual, Mirco Ravanelli, Joan Serrà, Antonio Bonafonte, and Yoshua Bengio. Learning problem-agnostic speech representations from multiple self-supervised tasks. In *Annual Conference of the International Speech Communication (Interspeech)*, pages 161–165, 2019.

[18] Anurag Kumar and Vamsi Krishna Ithapu. Secost: : Sequential co-supervision for large scale weakly labeled audio event detection. In *International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pages 666–670, 2020.

[19] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

[20] Daisuke Niizumi, Daiki Takeuchi, Yasunori Ohishi, Noboru Harada, and Kunio Kashino. Byol for audio: Self-supervised learning for general-purpose audio representation. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021.

[21] Amir Shirian and Tanaya Guha. Compact graph architecture for speech emotion recognition. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6284–6288, 2021.

[22] Amir Shirian, Subarna Tripathi, and Tanaya Guha. Dynamic emotion modeling with learnable graphs and graph inception network. *IEEE Transactions on Multimedia*, 2021.

[23] Taichi Ishiwatari, Yuki Yasuda, Taro Miyazaki, and Jun Goto. Relation-aware graph attention networks with relational position encodings for emotion recognition in conversations. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7360–7370, 2020.

[24] Jiawang Liu and Haoxiang Wang. Graph isomorphism network for speech emotion recognition. In *Annual Conference of the International Speech Communication (Interspeech)*, pages 3405–3409, 2021.

[25] Panagiotis Tzirakis, Anurag Kumar, and Jacob Donley. Multi-channel speech enhancement using graph neural networks. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3415–3419, 2021.

[26] Weizhi Nie, Minjie Ren, Jie Nie, and Sicheng Zhao. C-GCN: Correlation based graph convolutional network for audio-video emotion recognition. *IEEE Transactions on Multimedia*, 2020.

[27] Shuyun Tang, Zhaojie Luo, Guoshun Nan, Yuichiro Yoshikawa, and Ishiguro Hiroshi. Fusion with hierarchical graphs for mulitmodal emotion recognition. *arXiv preprint arXiv:2109.07149*, 2021.

[28] Yiwei Sun and Shabnam Ghaffarzadegan. An ontology-aware framework for audio event classification. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 321–325, 2020.

[29] Helin Wang, Yuexian Zou, Dading Chong, and Wenwu Wang. Modeling label dependencies for audio tagging with graph convolutional network. *IEEE Signal Processing Letters*, 27:1560–1564, 2020.

[30] Harsh Shrivastava, Yfang Yin, Rajiv Ratn Shah, and Roger Zimmermann. Mt-gcn for multi-label audio-tagging with noisy labels. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 136–140, 2020.

[31] Emily Alsentzer, Samuel G. Finlayson, Michelle M. Li, and Marinka Zitnik. Subgraph neural networks. In *NeurIPS*, 2020.

[32] Weilin Cong, Rana Forsati, Mahmut Kandemir, and Mehrdad Mahdavi. Minimal variance sampling with provable guarantees for fast training of graph neural networks. In *International Conference on Knowledge Discovery & Data Mining (ICDM)*, pages 1393–1403, 2020.

[33] Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. Graphsaint: Graph sampling based inductive learning method. In *International Conference on Learning Representations (ICLR)*, 2020.

[34] Kexin Huang and Marinka Zitnik. Graph meta learning via local subgraphs. In *NeurIPS*, 2020.

[35] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, pages 1024–1034, 2017.

[36] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori. Speech enhancement based on deep denoising autoencoder. In *Annual Conference of the International Speech Communication (Interspeech)*, pages 436–440, 2013.

[37] Bahare Fatemi, Layla El Asri, and Seyed Mehran Kazemi. Slaps: Self-supervision improves structure learning for graph neural networks. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

[38] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780, 2017.

[39] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. CNN architectures for large-scale audio classification. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 131–135, 2017.

[40] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[42] Yuan Gong, Yu-An Chung, and James Glass. AST: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*, 2021.

[43] Wei Dai, Chia Dai, Shuhui Qu, Juncheng Li, and Samarjit Das. Very deep convolutional neural networks for raw waveforms. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 421–425, 2017.

[44] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D Plumbley. PANNs: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894, 2020.

[45] Hassan Akbari, Linagzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. VATT: Transformers for multimodal self-supervised learning from raw video, audio and text. *arXiv preprint arXiv:2104.11178*, 2021.

[46] Jian Huang, Ya Li, Jianhua Tao, Zheng Lian, Mingyue Niu, and Jiangyan Yi. Speech emotion recognition using semi-supervised learning with ladder networks. In *Asian Conference on Affective Computing and Intelligent Interaction*, pages 1–5, 2018.

[47] Jingjun Liang, Ruichen Li, and Qin Jin. Semi-supervised multi-modal emotion recognition with cross-modal distribution matching. In *ACM International Conference on Multimedia*, pages 2852–2861, 2020.

[48] Dongwei Jiang, Wubo Li, Miao Cao, Ruixiong Zhang, Wei Zou, Kun Han, and Xiangang Li. Speech SIMCLR: Combining contrastive and reconstruction objective for self-supervised speech representation learning. *arXiv preprint arXiv:2010.13991*, 2020.

[49] Yuan Gong, Cheng-I Jeff Lai, Yu-An Chung, and James Glass. Ssast: Self-supervised audio spectrogram transformer. *arXiv preprint arXiv:2110.09784*, 2021.

[50] Neil Scheidwasser-Clow, Mikolaj Kegler, Pierre Beckmann, and Milos Cernak. Serab: A multi-lingual benchmark for speech emotion recognition. *arXiv preprint arXiv:2110.03414*, 2021.

[51] Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y Lin, Andy T Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, et al. Superb: Speech processing universal performance benchmark. *arXiv preprint arXiv:2105.01051*, 2021.

[52] S Mao, PC Ching, and T Lee. Deep learning of segment-level feature representation with multiple instance learning for utterance-level speech emotion recognition. In *Annual Conference of the International Speech Communication (Interspeech)*, pages 1686–1690, 2019.

[53] Bo-Hao Su, Chun-Min Chang, Yun-Shao Lin, and Chi-Chun Lee. Improving speech emotion recognition using graph attentive bi-directional gated recurrent unit network. In *Annual Conference of the International Speech Communication (Interspeech)*, pages 506–510, 2020.

[54] Shuiyang Mao, P. C. Ching, C.-C. Jay Kuo, and Tan Lee. Advancing multiple instance learning with attention modeling for categorical speech emotion recognition. In *Annual Conference of the International Speech Communication (Interspeech)*, 2020.

[55] Lu Yi and Man-Wai Mak. Improving speech emotion recognition with adversarial data augmentation network. *IEEE Trans on Neural Networks and Learning Systems*, 2020.

[56] C Busso, M Bulut, C-C Lee, A Kazemzadeh, E Mower, S Kim, J N Chang, S Lee, and S S Narayanan. IEMOCAP: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335, 2008.

[57] C Busso, S Parthasarathy, A Burmania, M AbdelWahab, N Sadoughi, and E M Provost. MSP-IMPROV: An acted corpus of dyadic interactions to study emotion perception. *IEEE Transactions on Affective Computing*, 8(1):67–80, 2016.

[58] B Schuller, S Steidl, and A Batliner. The interspeech2009 emotion challenge. In *Annual Conference of the International Speech Communication (Interspeech)*, 2009.

[59] F Eyben, F Weninger, F Gross, and B Schuller. Recent developments in opensmile, the munich open-source multimedia feature extractor. In *ACM International Conference on Multimedia*, pages 835–838, 2013.

[60] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–93, 2008.

[61] Galileo Namata, Ben London, Lise Getoor, and Bert Huang. Query-driven active surveying for collective classification. In *International Workshop on Mining and Learning with Graphs*, volume 8, 2012.

[62] Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*, 2020.

[63] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[64] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019.

[65] Vikas Verma, Meng Qu, Kenji Kawaguchi, Alex Lamb, Yoshua Bengio, Juho Kannala, and Jian Tang. Graphmix: Improved training of GNNs for semi-supervised learning. In *AAAI Conference on Artificial Intelligence*, volume 35, pages 10024–10032, 2021.

[66] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *AAAI Conference on Artificial Intelligence*, volume 34, pages 3438–3445, 2020.

[67] Anurendra Kumar, Tanaya Guha, and Prasanta Kumar Ghosh. Dirichlet latent variable model: A dynamic model based on Dirichlet prior for audio processing. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(5):919–931, 2019.

Amir Shirian is currently a PhD student in the Department of Computer Science at the University of Warwick, UK. He has received his BSc (2015) and MSc (2018) degrees in Electrical Engineering from the University of Tehran, Iran. His research interests include multimodal signal processing, graph neural networks, and machine learning with applications to emotion and behaviour understanding.

Krishna Somandepalli received his PhD in Electrical and Computer Engineering from the University of Southern California, CA, USA and a Masters degree from the University of California at Santa Barbara, CA, USA in Electrical and Computer Engineering. Following his Masters degree, he worked as an assistant research scientist at NYU Langone medical Center, New York, NY, USA. He currently works at Google Research. His research interests include multimodal machine learning, media understanding and developing inclusive technologies.

Tanaya Guha has received her PhD in Electrical and Computer Engineering from the University of British Columbia, Vancouver, Canada. She is currently a Senior Lecturer (Associate Professor) in the School of Computing Science, University of Glasgow, UK. Her research is focused on modelling and analysis of audio and visual data combining machine learning and signal processing with applications in healthcare and autonomous systems. She is a member of the Editorial Boards of Nature Scientific Reports and APSIPA Transactions on Signal and Information Processing. She is an elected member of IEEE Multimedia Systems Applications Technical Committee. She regularly serves in the organizing and program committees of ICME, ICMI, WACV and INTERSPEECH.