Graph-based Representation of Audio signals for Sound Event Classification

Carlo Aironi, *Samuele Cornell*, *Emanuele Principi*, *Stefano Squartini*
Department of Information Engineering, Università Politecnica delle Marche, Italy
Email: c.aironi@pm.univpm.it, s.cornell@pm.univpm.it, e.principi@univpm.it, s.squartini@univpm.it

***Abstract*—In recent years there has been a considerable rise in interest towards Graph Representation and Learning techniques, especially in such cases where data has intrinsically a graph-like structure: social networks, molecular lattices, or semantic interactions, just to name a few. In this paper, we propose a novel way to represent an audio signal from its spectrogram by deriving a graph-based representation which can be then employed by already established Graph Deep-Neural-Networks techniques. We evaluate this approach on a Sound Event Classification task by employing the widely used ESC and Urbansound8k datasets and compare it with a Convolutional Neural Network (CNN) based method. We show that such proposed graph-based approach is extremely compact and used in conjunction learned CNN features, allows for a significant increase in classification accuracy over the baseline with more than 50 times less parameters than the original CNN method. This suggests that, the proposed graph-based features can offer additional discriminative information on top of learned CNN features.**

***Index Terms*—Sound Event Classification, Graph Representation Learning, Graph Neural Networks, Convolutional Neural Networks.**

I. Introduction

Sound Event Classification (SEC) consists in the automatic recognition of different sound events and has a wide range of applications among different domains, ranging from Autonomous driving, wearable devices, Human-Computer Interfaces for people with hearing impairments [1], home automation [2], surveillance systems [3], hazardous environment monitoring or as a part of Acoustic Scene Recognition [4].

The SEC problem has been historically first approached with classical machine learning algorithms like Gaussian Mixture Models (GMM) [5], Support Vector Machines [6], Hidden Markov Models [7], using handcrafted features, such as MFCC coefficients [8], Mel and log-Mel filterbank features, gammatone coefficients [9] and wavelet features [10].

A significant growth in automatic recognition accuracy has been achieved by using Deep Neural Network (DNN) based methods. Among these methods the most successful use Convolutional Neural Networks (CNN) [11]–[14] classifiers, on either spectrogram-based features [11]–[13] or directly from the raw waveform [14].

In this paper, we propose a novel graph-based representation of an audio signal inspired by recent successes in image classification [15], [16]. Hereafter, we present a radically different method from [15], [16] which allows to extract a graph-based representation from an audio signal enabling the use of Graph-Neural-Networks (GNNs) methods.

Starting from the log-scaled Mel spectrogram we propose to build a graph with heuristic rules defining its basic elements: Nodes, Edges and corresponding Attribute vectors. This representation allows us to apply GNN-based supervised learning methods, which have been introduced recently for inherently graph-structured data such as social network interactions [17] and Chemical graph-structured data [18]. To the authors knowledge, this is the first work in which a graph-based representation together with GNNs are applied in the audio classification domain.

This paper is organized as follows: in Section I-A GNNs and graphs are briefly introduced. Following, in Section II we explain in detail the proposed approach, and then in Section III and IV we describe respectively the datasets used and the neural architectures employed in the experiments. We present and discuss the experimental results in V and finally in VI we draw conclusions and outline possible future work.

A. Graphs and Graph Neural Networks

Graphs are data structures widely used in many scientific fields due to their ability to provide a "natural" way to represent information in many contexts.

Basic elements of a graph, Nodes and Edges (interconnections) are defined heuristically from prior knowledge of the problem faced, and can strongly depend on its domain. However, the absence of an Euclidean structure makes graphs challenging to process using DNNs in a conventional way.

For this reason, specific Learning-on-graph techniques have been developed, historically with the aim to generate knowledge models, extracting informations from node attributes, edge attributes and graph topology [19]. Early studies on this field involve the application of recurrent structures on acyclic directed graphs and fall into the category of recurrent graph neural networks (RecGNNs [20]). Subsequently a large number of methods were developed to formalize the concept of graph convolution: spectral-based convolutional GNNs (ConvGNNs [21]) use the Fourier transform on the graph Laplacian Matrix and spatial-based ConvGNNs [18] exploit topological connections through the *message passing* mechanism between nodes. Further developments led to Graph Autoencoders GAEs [22], unsupervised methods for graph generation and embedding, and Spatial-Temporal Graph Neural Networks (STGNNs [23]) for time dynamic graphs. Typical problems addressed by GNNs fall into the categories of Node classification, Graph

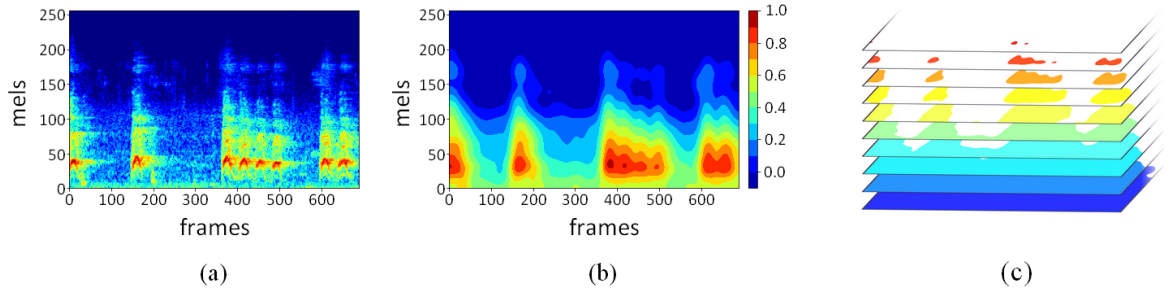

Fig. 1. Log-Mel spectrogram of an audio clip from Urbansound8k dataset, "dog bark" category, (a), segmented version (b) and exploded view (c).

classification, Node prediction, Link prediction, Clustering and Node/Graph similarity detection.

II. Proposed Method

In the proposed method we build a graph whose informative elements (nodes, edges and their attributes) are derived from the log-Mel scale spectrogram of a signal using an image processing approach. A segmentation procedure is performed over the log-Mel representation to isolate several different graphical entities we call *regions*. Each of these entities are then associated with a node in the graph. In our preliminary experiments we found out that most commonly segmentation methods used in image processing (like superpixel segmentation with N-cut [24] or SLIC [25] algorithm) perform very poorly if applied directly to spectrograms, due to the lack of both color depth and sharp edges. We instead use a Level-Set method [26] to define level curves which enclose *regions* with constant acoustic energy. In order to reduce the amount of resulting *regions*, and thus the number of nodes, before the segmentation step a 2D Gaussian filter with squared kernel is used to obtain a smoothed version of the log-Mel spectra. After smoothing the log-Mel spectrogram range is normalized to $[0, 1]$ range. Figure 1 shows a log-Mel scale spectrogram belonging to a clip extracted from Urbansound8k dataset (a) and its smoothed and segmented version, (b) and (c).

We denote with $x(t, f)$ the normalized and smoothed log-Mel spectra, where $0 \leq f < F$ and $0 \leq t < T$ are the Mel band and frame indexes and F and T are the total number of Mel bands and frames. *Regions* are extracted by considering a finite set of K thresholds $\tau = [\tau_1, \ldots, \tau_K]$, with each threshold $\tau_i \in (0, 1)$. Level-sets are then extracted applying a function to the normalized log-Mel spectrogram defined as follows:

$$f\left(x(t,f), \tau_i\right) = \begin{cases} 0 & \text{if } x(t,f) \leq \tau_i \\ 1 & \text{if } x(t,f) > \tau_i \end{cases}. \tag{1}$$

This step returns a set of $K + 1$ discrete *levels* as it can be seen in Figure 1 (c) where, for example, 10 *levels* in which the acoustic energy falls above a certain threshold are identified.

As defined in Equation 1, each log-Mel spectra *level* is a binary matrix, it is thus possible to apply a trivial image segmentation procedure to isolate *regions* for each *level*. In detail, we can isolate each *region* $I\left(t; f\right)$, by taking each maximally contiguous area where $f(x(t, f), \tau_i) = 1$; in Figure 1 (c) for example, there are 4 different regions in the third *level* from the top. This segmentation procedure can be implemented very efficiently using dynamic programming.

A. Node Attributes and Graph Edges

Regions arising from the segmentation step are assigned to graph Nodes which are characterized through attributes encoding each *region* geometric shape and position. These attributes are derived from the ith and jth order image moments $M_{i,j}$, central moments $\mu_{i,j}$, and covariance matrix $cov\left[I\left(t; f\right)\right]$ of the *region* $I\left(t; f\right)$. The ith and jth order image moments are defined as:

$$M_{ij} = \sum_t \sum_f t^i f^j \cdot I\left(t; f\right), \tag{2}$$

where, as before, t and f denote the frame and Mel band indexes (x-y image coordinates), while the corresponding central moments:

$$\mu_{ij} = \sum_t \sum_f \left(t - t_c\right)^i \left(f - f_c\right)^j \cdot I\left(t; f\right), \tag{3}$$

where t_c and f_c are the *region* centroid along frame axis and Mel band axis, defined as:

$$t_c = \frac{M_{10}}{M_{00}},\ f_c = \frac{M_{01}}{M_{00}}. \tag{4}$$

The covariance matrix is obtained from the central moments:

$$cov\left[I\left(t; f\right)\right] = \begin{bmatrix} \mu_{20}/\mu_{00} & \mu_{11}/\mu_{00} \\ \mu_{11}/\mu_{00} & \mu_{02}/\mu_{00} \end{bmatrix} = \begin{bmatrix} \mu'_{20} & \mu'_{11} \\ \mu'_{11} & \mu'_{02}. \end{bmatrix} \tag{5}$$

In this work we use eight attributes defined as follows:

- Area (which corresponds to moment M_{00}).
- Perimeter (corresponding to the moment M_{00} of the region boundary).
- The centroid spatial coordinates, t_c, f_c, and z_c (along *level* axis).
- Orientation, defined as the angle θ between major axis and the vertical axis of an ellipse with the same image moment of the region. It can be obtained from the covariance matrix elements:

$$\theta = \frac{1}{2} arctan\left(\frac{2\mu'_{11}}{\mu'_{20} - \mu'_{02}}\right) \quad (6)$$

- Eccentricity *E*, defined as the ratio between focal distance and the semi-major axis of an ellipse with the same image moment of the region. It can be obtained from the eigenvalues λ_i of the covariance matrix:

$$E = \sqrt{1 - \frac{\lambda_{min}}{\lambda_{max}}} \quad (7)$$

- Solidity, which encodes if the shape is convex or concave and is defined as the ratio between the area of the region and the area of a convex hull, the smallest polygon enclosing the region.

Edges of graph are defined by the following empirical rule: two nodes i, j, each corresponding to *regions* $I_i(t, f)$ and $I_j(t, f)$ are connected if they intersect $I_i(t, f) \cap I_j(t, f) \neq \emptyset$. Edges orientation are based on the relative *levels* of the two *regions*: from lower to higher. A directed graph is thus obtained. Other criteria to define edges were explored but led to worse performance.

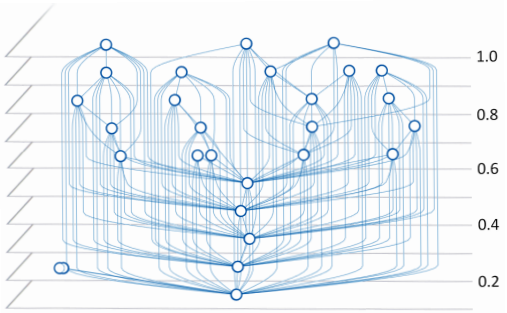

Fig. 2. Graph originated from Fig. 1 spectrogram

III. Datasets

We evaluate the performance of the proposed method using two datasets widely employed for SEC: ESC10 which is a subset of the wider ESC50 collection [27], and Urbansound8k [28]. We describe them thereafter.

A. ESC10

ESC10 consists in 400 audio clips, grouped in 10 classes. Each clip has a duration of 5 seconds and is sampled at 44100 Hz. Due to the scarcity of audio clips we applied the same data augmentation technique to be directly comparable with [27], using *pitch shift*, *time stretch* and *time shift* transformations.

B. Urbansound8k

Urbansound8k is composed by 8732 registrations of urban environmental sounds, grouped in 10 classes. Clips on Urbansound8k have different lengths and different sampling frequencies, so a resampling to 22050 Hz and a zero padding in time is performed.

IV. Experimental Setup

To evaluate the efficacy of the proposed graph-based representation we considered two different DNN classification approaches for each dataset:

- Graph-only (GNN), in which only the proposed graph-based features are used and a GNN is employed for classification.
- Hybrid (GNN+CNN), in which the graph-only approach is combined using a stacking ensemble approach with more standard CNN-based features extracted from log-Mels. The two high-level features are then combined using a Multi-Layer-Perceptron (MLP).

We compare these two approaches with a state-of-the-art CNN-based architecture proposed in [11]. More in detail, we use the short-segment majority voting architecture from this latter work as our baseline system (CNN) as well as for the Hybrid (GNN+CNN) approach. This model takes in input the log-Mel spectra of the audio signal and processes it with a cascade of 2D convolutional layers with Rectified Linear Unit (ReLU) activations followed by two fully connected layers with ReLU non-linearity and an output linear layer. 60 Log-Mel bands are employed with a window of 1024 samples and 50% overlap. All networks are trained to convergence by using early stopping and halving the learning rate if no improvement is observed for 5 epochs.

A. Graph-only

The GNN employed in this work belongs to the category of Message Passing Neural Networks (MPNN) [18], [29] which are composed of several graph convolutional layers (GNNConv). The architecture is depicted in Figure 3. Each of the GNNConv layers transform the input graph into another one with same topological structure but whose nodes have an higher dimensional feature vector. GNNConv is defined as:

$$x'_i = \mathbf{\Theta} x_i + \frac{1}{|N(i)|} \sum_{j \in N(i)} x_j \cdot h_{\mathbf{\Gamma}}(e_{i,j}) \quad (8)$$

where x_i is the input node, $\mathbf{\Theta}$ is a learnable $I \times C$ linear transformation, and $h_{\mathbf{\Gamma}}$ is a non-linear transformation with $\mathbf{\Gamma}$ learnable parameters (here we use a MLP with ReLU), fed with the node distances $e_{i,j}$ between node *i* and its neighboring nodes $N(i)$.

After every GNNConv layer, an Exponential Linear Unit (ELU) activation is applied and the graph is shrunk through a graph-level pooling phase applied on clusters of two nodes, as paired by the Graclus algorithm [30]. The last layer rejects the informative content of the graph structure by keeping only node attributes (node-level pooling) which are then embedded in a single vector describing the whole graph (global-pooling). Finally, a MLP with one hidden layer and ReLU activation, is used to obtain class logits.

We used PyTorch Geometric library [31] for the implementation. Each GNNConv layers has 32 channels C for $\mathbf{\Theta}$ and each MLP has an hidden size of 64. The readout MLP has an hidden size of 128. We use for training Adam optimizer [32]

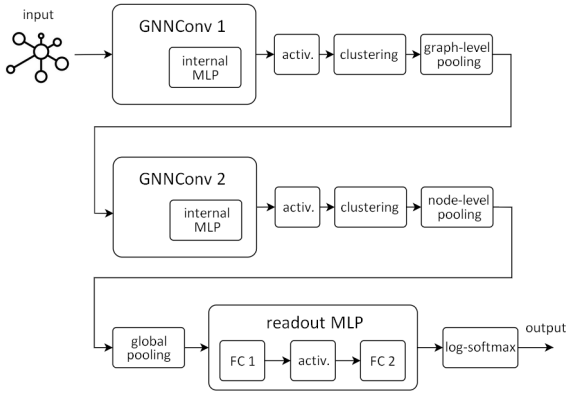

Fig. 3. GNN architecture used in both Graph-only and Hybrid approaches.

with a batch size of 32 and a learning rate of 0.001. For ESC10 we use 2 layers of GNNConv and 3 layers for Urbansound8k.

B. Hybrid

In the Hybrid approach we combine the high-level graph-based features extracted with the MPNN described in previous Section, with high-level CNN extracted features in a stacking ensemble fashion. Regarding the CNN, we use the same architecture as in [11], which is also our baseline model.

Here, we consider for this hybrid approach only the top convolutional layers of the baseline CNN architecture [11] and concatenate the embeddings as extracted from such layers with the one obtained by the MPNN before the readout MLP. This hybrid representation is fed to an MLP and then to a linear output layer which outputs class logits. This architecture is depicted in Figure 4.

We re-use the pre-trained MPNN from Graph-only approach as well as pre-trained CNN layers obtained by a re-implementation of the network from [11]. In the training phase only the fusion MLP and the output layer are updated, the CNN and GNN branches are kept freezed. We use Stochastic Gradient Descent optimizer with Nesterov momentum, batch size 64 and learning rate of 0.001. The fusion MLP has 1024 hidden neurons and ReLU activation.

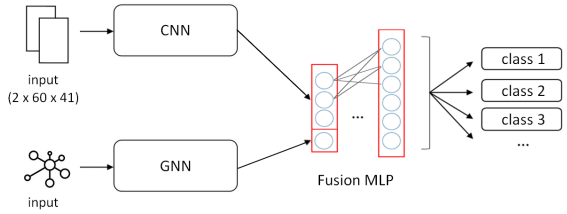

Fig. 4. Hybrid GNN-CNN ensembling scheme

V. Results

In the following, we report and discuss the results obtained by the previously defined Graph-only and Hybrid classifiers. To be comparable with [11], we calculate Accuracy by using 5-fold cross-validation for ESC10 and 10-fold cross-validation for Urbansound8k. In both cases, the fold partitions are the same defined by the dataset guidelines. We give the neural networks trainable parameters counts in Table I and report Accuracy in Table II as well as in Figure 5 were we show box-plots. In Table II we highlight in bold best results validated through a Paired Student t-test [33] with a confidence level of 95% performed on 10 different runs (10 different folds for Urbansound8k and 10 for ESC10).

TABLE I
Total learnable parameters for different structures

	ESC10	Urbansound8k
CNN	26M	26M
GNN	84K	152K
GNN+CNN	**369K**	**437K**

It can be observed that the Graph-only approach (GNN) has overall less accuracy than the CNN-based approach, with a significant difference especially for Urbansound8k which is more noisy. On the other hand, this classifier has more than 100 times less the number of parameters and, moreover the size of proposed graph-based features is significantly lower than the log-Mel features. In fact, for an ESC10 audio clip the full size of log-Mels as employed in [11] is 25840 while for the proposed graph-based features only 30 nodes (on average on ESC10) are extracted, with each node having 8 scalar features as described in Section II. Thus the proposed representation is extremely compact and this can explain the difference in performance.

Nonetheless, these very compact features are able to bring considerable improvement when combined with the CNN features in the Hybrid (GNN+CNN) approach. This Hybrid model is still considerably smaller than the CNN baseline due to the use of a small fusion MLP. This result suggests that the proposed graph-based approach is able to supply additional discriminative information with respect to CNN learned features, despite the modest size of proposed representation.

TABLE II
Overall classification Accuracy for 5-folds (ESC-10) and 10-folds (Urbansound8k).

Method	ESC10	Urbansound8k
CNN [11]	0.775	0.700
GNN	0.737	0.635
GNN+CNN	**0.800**	**0.730**

VI. Conclusions

In this work we presented a novel method which allows to represent the information contained in the log-scaled Mel spectrograms through a graph using a segmentation step based on constant energy level curves and image processing techniques. This graph-based representation is remarkably dense and suitable for resource constrained device and edge-computing devices. The proposed approach is applied to a

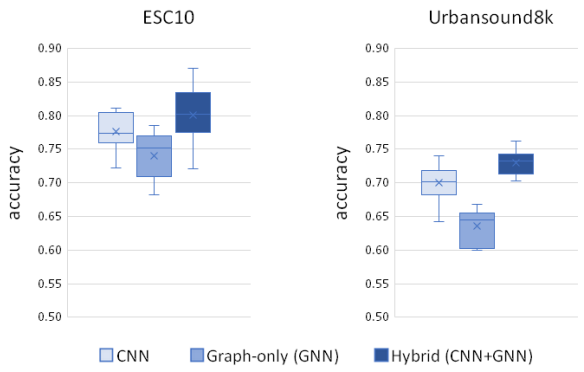

Fig. 5. Box plots for classification Accuracy for the baseline (CNN) and the proposed configurations (GNN, GNN+CNN), for ESC10 dataset (left) and Urbansound8k dataset (right).

Sound Event Classification task using two real-world datasets and compared with a state-of-the-art CNN based model.

We found that although the proposed graph-based representation is not able to compete with current state-of-the-art CNN-based models, due to its modest size, it is able to offer additional discriminative capability when used in conjunction with standard CNN learned features, significantly boosting performance and allowing to reduce drastically the size of the network.

Future work includes exploring different GNN models that could potentially further improve both the computational footprint and performance as well as devising a method for learning to extract the graph-based representation without relying on any a-priori assumption.

REFERENCES

[1] R. Lyon, "Machine hearing: An emerging field," *Signal Processing Magazine, IEEE*, vol. 27, pp. 131 – 139, 2010.

[2] M. Vacher, J.-F. Serignat, and S. Chaillol, "Sound classification in a smart room environment: an approach using gmm and hmm methods," in *IEEE Conference on Speech Technology and Human-Computer Dialogue*, vol. 1, 2007, pp. 135–146.

[3] R. Radhakrishnan, A. Divakaran, and A. Smaragdis, "Audio analysis for surveillance applications," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2005, pp. 158–161.

[4] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, pp. 16–34, 2015.

[5] T. Virtanen, A. Mesaros, T. Heittola, M. Plumbley, P. Foster, E. Benetos, and M. Lagrange, *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*. Tampere University of Technology. Department of Signal Processing, 2016.

[6] B. Uzkent, B. Barkana, and H. Cevikalp, "Non-speech environmental sound classification using svms with a new set of features," *International Journal of Innovative Computing, Information and Control*, vol. 8, 05 2012.

[7] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real-life recordings," 07 2014.

[8] S. Chu, S. Narayanan, and C. . J. Kuo, "Environmental sound recognition with time–frequency audio features," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1142–1158, 2009.

[9] X. Valero and F. Alías, "Gammatone cepstral coefficients: Biologically inspired features for non-speech audio classification," *Multimedia, IEEE Transactions on*, vol. 14, pp. 1684–1689, 2012.

[10] J. T. Geiger and K. Helwani, "Improving event detection for audio surveillance using gabor filterbank features," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, 2015, pp. 714–718.

[11] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2015, pp. 1–6.

[12] J. Salamon and J. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. PP, 01 2017.

[13] H. Zhang, I. McLoughlin, and Y. Song, "Robust sound event recognition using convolutional neural networks," in *ICASSP*, 2015, pp. 559–563.

[14] S. Abdoli, P. Cardinal, and A. L. Koerich, "End-to-end environmental sound classification using a 1d convolutional neural network," *Expert Systems with Applications*, vol. 136, pp. 252–263, 2019.

[15] B. Knyazev, X. Lin, M. R. Amer, and G. W. Taylor, "Image Classification with Hierarchical Multigraph Networks," *arXiv e-prints*, p. arXiv:1907.09000, 2019.

[16] A. Quek, Z. Wang, J. Zhang, and D. Feng, "Structural image classification with graph neural networks," in *International Conference on Digital Image Computing: Techniques and Applications*, 2011, pp. 416–421.

[17] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *The World Wide Web Conference*, 2019, p. 417–426.

[18] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1263–1272.

[19] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *arXiv preprint arXiv:1709.05584*, 2017.

[20] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *Trans. Neur. Netw.*, p. 61–80, 2009.

[21] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *ArXiv*, 2015.

[22] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *stat*, vol. 1050, p. 21, 2016.

[23] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting," in *International Joint Conference on Artificial Intelligence*, 2018, pp. 3634–3640.

[24] K. Ersahin, "Segmentation and classification of polarimetric sar data using spectral graph partitioning," Ph.D. dissertation, 2009.

[25] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[26] N. D. Katopodes, *Free-Surface Flow: computational methods*. Elsevier, 2019.

[27] K. J. Piczak, "Esc: Dataset for environmental sound classification," in *Proceedings of the 23rd ACM International Conference on Multimedia*, 2015, p. 1015–1018.

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

[29] M. Simonovsky and N. Komodakis, "Dynamic edge-conditioned filters in convolutional neural networks on graphs," *CoRR*, vol. abs/1704.02901, 2017.

[30] I. S. Dhillon, Y. Guan, and B. Kulis, "Weighted graph cuts without eigenvectors a multilevel approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 11, pp. 1944–1957, 2007.

[31] M. Fey and J. E. Lenssen, "Fast graph representation learning with pytorch geometric," *CoRR*, vol. abs/1903.02428, 2019.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[33] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, pp. 1895–1923, 1998.