# GEOMETRIC SCATTERING ATTENTION NETWORKS

*Yimeng Min*$^{1,3,\star}$ *Frederik Wenkel*$^{2,3,\star}$ *Guy Wolf*$^{2,3}$

Université de Montréal, [1]Department of Computer Science & Operational Research
[2]Department of Mathematics & Statistics; [3]Mila – Quebec AI Institute, Montreal, QC, Canada

## ABSTRACT

Geometric scattering has recently gained recognition in graph representation learning, and recent work has shown that integrating scattering features in graph convolution networks (GCNs) can alleviate the typical oversmoothing of features in node representation learning. However, scattering methods often rely on handcrafted design, requiring careful selection of frequency bands via a cascade of wavelet transforms, as well as an effective weight sharing scheme to combine together low- and band-pass information. Here, we introduce a new attention-based architecture to produce adaptive task-driven node representations by implicitly learning node-wise weights for combining multiple scattering and GCN channels in the network. We show the resulting geometric scattering attention network (GSAN) outperforms previous networks in semi-supervised node classification, while also enabling a spectral study of extracted information by examining node-wise attention weights.

***Index Terms*—** Graph neural networks, geometric scattering, attention, node classification, geometric deep learning

## 1. INTRODUCTION

Convolutional neural networks (CNNs) have shown great success on a range of tasks, including image classification, machine translation and speech recognition. By optimizing the local filters within a neural network-based architecture, the model is able to construct expressive representations and thus tackle tasks on regular Euclidean data. Based on CNNs, graph neural networks (GNNs) [1–4] show pronounced performance on non-Euclidean data such as molecule modelling and node classification tasks. The generalization from regular grids to irregular domains is usually implemented under the spectral graph theory framework [5,6]. Specifically, GNNs learn a state embedding with aggregated neighbourhood information using a one-step neighbourhood-localized filter, which corresponds to the lowest frequency of a graph Laplacian.

However, the introduction of localized filter implements local smoothing of the input message. Previous studies suggest the convolutional operation performs low-pass filtering on the feature vectors [7, 8], which forces a smooth embedding of neighbouring nodes, and leads to information loss in message passing and severely degrading their performance [9, 10]. To assign different weights to the neighbouring nodes, graph attention networks (GATs) leverage attention layers to learn the adaptive weights across the edges, resulting in a leap in model capacity [3]. While the attention layers increase the model interpretability, the resulting networks still rely on averages of neighboring node features for similarity computations, rather than more complex regularity patterns.

In order to capture higher-order regularity on graphs, geometric scattering networks were recently introduced [11–14]. These networks generalize the Euclidean scattering transform [15–17] to the graph domain and leverage graph wavelets to extract effective and efficient graph representations. In [18], a hybrid scattering graph convolutional network (Sc-GCN) is proposed in order to tackle oversmoothing problems in traditional GCNs. Geometric scattering together with GCN-based filters are used to apply both band-pass and low-pass filters to the graph signal. However, even though Sc-GCN achieves good perfomance on a range of node level classification tasks, it requires the selection of a task-appropriate configuration of the network and its scattering wavelet composition.

Here, we introduce a geometric scattering attention network (GSAN) that combines the hybrid Sc-GCN approach together with a node-wise attention mechanism to automatically adapt its filter (or *channel*) composition, thus simplifying its architecture tuning. We evaluate our proposed approach on a variety of semi-supervised node classification benchmarks, demonstrating its performance improvement over previous GNNs. Finally, we also show that visualizing the distribution of node-wise attention weights enables further analysis of task-dependent information extracted from node features.

## 2. PRELIMINARIES

We consider a weighted graph $G = (V, E, w)$ with nodes $V := \{v_1, \ldots, v_n\}$ and (undirected) edges $E \subset \{\{v_i, v_j\} \in V \times V, i \neq j\}$. The function $w : E \to (0, \infty)$ assigns positive weights to the graph edges, which we aggregate into the

$^\star$ Equal contribution; student authors; order determined alphabetically. Correspondence to guy.wolf@umontreal.ca . This work was partially funded by IVADO Professor startup & operational funds, IVADO Fundamental Research Proj. grant PRF-2019-3583139727, and NIH grant R01GM135929. The content provided here is solely the responsibility of the authors and does not necessarily represent the official views of the funding agencies.

*adjacency matrix* $\boldsymbol{W} \in \mathbb{R}^{n \times n}$ via

$$\boldsymbol{W}[v_i, v_j] := \begin{cases} w(v_i, v_j) & \text{if } \{v_i, v_j\} \in E, \\ 0 & \text{otherwise.} \end{cases}$$

Each node $v_i \in V$ further possesses a feature vector $\boldsymbol{x}_i \in \mathbb{R}^{d_0}$, which we aggregate into the feature matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d_0}$. We further define the *degree matrix* $\boldsymbol{D} \in \mathbb{R}^{n \times n}$, defined by $\boldsymbol{D} := \boldsymbol{D}(\boldsymbol{W}) := \operatorname{diag}(d_1, \ldots, d_n)$ with $d_i := \deg(v_i) := \sum_{j=1}^{n} \boldsymbol{W}[v_i, v_j]$ being the *degree* of the node $v_i$. The GNN methods discussed in the following yield layer-wise node representations, compactly written as $\boldsymbol{H}^\ell \in \mathbb{R}^{n \times d_\ell}$ for the $\ell^{th}$ layer with $\boldsymbol{H}^0 := \boldsymbol{X}$.

### 2.1. Graph Convolutional Networks

A very popular method introduced in [2] connects the local node-based information in $\boldsymbol{X}$ with the intrinsic data-geometry encoded by $\boldsymbol{W}$. This is realized by filtering the node features with the layer-wise update rule

$$\boldsymbol{H}^\ell = \sigma(\boldsymbol{A}\boldsymbol{H}^{\ell-1}\boldsymbol{\Theta}^\ell). \tag{1}$$

The matrix multiplication with

$$\boldsymbol{A} := (\boldsymbol{D} + \boldsymbol{I}_n)^{-1/2} (\boldsymbol{W} + \boldsymbol{I}_n) (\boldsymbol{D} + \boldsymbol{I}_n)^{-1/2}$$

constitutes the filtering operation, while the multiplication with $\boldsymbol{\Theta}^\ell$ followed by an elementwise nonlinerity $\sigma(.)$ can be seen as a linear layer applied to the filtered node features.

This method is subject to the so-called oversmooting problem [9], which causes the node features to be smoothed out, the more GCN layers are iterated. In signal processing terminology, the update rule can be seen as a low-pass filtering operation [18] so that the model cannot access a significant potion of the information considered in the frequency domain.

### 2.2. Geometric Scattering

Recently, geometric scattering was introduced to incorporate band-pass filters in GNNs [11, 18], inspired by the utilization of scattering features in the analysis of images [16, 19] and audio signals [17, 20]. There, cascades of wavelets can often recover high-frequency information and geometric scattering exhibits an analogous property on graph domains.

Geometric scattering is based on the lazy random walk matrix $\boldsymbol{P} := \frac{1}{2}(\boldsymbol{I}_n + \boldsymbol{W}\boldsymbol{D}^{-1})$, which is used to construct diffusion wavelet matrices $\boldsymbol{\Psi}_k \in \mathbb{R}^{n \times n}$ [21] of scale $2^k$,

$$\begin{cases} \boldsymbol{\Psi}_0 := \boldsymbol{I}_n - \boldsymbol{P}, \\ \boldsymbol{\Psi}_k := \boldsymbol{P}^{2^{k-1}} - \boldsymbol{P}^{2^k}, \quad k \geq 1. \end{cases} \tag{2}$$

For node features $\boldsymbol{X}$, the scattering features are calculated as

$$\boldsymbol{U}_p\boldsymbol{X} := \boldsymbol{\Psi}_{k_m} |\boldsymbol{\Psi}_{k_{m-1}} \ldots |\boldsymbol{\Psi}_{k_2} |\boldsymbol{\Psi}_{k_1} \boldsymbol{X}|| \ldots |,$$

with scattering scales $p := (k_1, \ldots, k_m) \in \cup_{m \in \mathbb{N}} \mathbb{N}_0^m$.

Similar to GCN layers, the scattering update rule has the form

$$\boldsymbol{H}^\ell := \sigma\left(\boldsymbol{U}_p \boldsymbol{H}^{\ell-1} \boldsymbol{\Theta}^\ell\right). \tag{3}$$

In [18], a hybrid architecture (referred to as Sc-GCN here) is proposed in order to combine the benefits of both GCN and scattering filters. Therefore, *network channels* $\{\boldsymbol{H}_i^\ell\}_{i=1}^m$, each coming from either GCN (Eq. 1) or scattering (Eq. 3), are concatenated (horizontally) constituting the *hybrid layer*

$$\boldsymbol{H}^\ell := \left[\boldsymbol{H}_1^\ell \parallel \ldots \parallel \boldsymbol{H}_m^\ell\right].$$

### 2.3. Graph Residual Convolution

This architecture component from [18] constitutes an adjustable low-pass filter, parameterized by the matrix

$$\boldsymbol{A}_{res}(\alpha) = \frac{1}{\alpha + 1}(\boldsymbol{I}_n + \alpha \boldsymbol{W}\boldsymbol{D}^{-1}),$$

which is usually applied to $\boldsymbol{H}^\ell$ followed by a fully connected layer (without nonlinearity). It filters the output of the hybrid layer for high-frequency noise, which can occur as a result of scattering features.

### 2.4. Graph Attention Networks

Another popular approach for node classification tasks was introduced in [3], where at any node $v_i$, an attention mechanism attends over the aggregation of node features from the node neighborhood $\mathcal{N}_i$. The aggregation coefficients are learned via

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\boldsymbol{a}^T [\boldsymbol{\Theta}\boldsymbol{h}_i \parallel \boldsymbol{\Theta}\boldsymbol{h}_j])}{\sum_{v_k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(\boldsymbol{a}^T [\boldsymbol{\Theta}\boldsymbol{h}_i \parallel \boldsymbol{\Theta}\boldsymbol{h}_k])}.$$

where $\boldsymbol{h}_i \in \mathbb{R}^d$ is the features of node $i$, $\boldsymbol{\Theta} \in \mathbb{R}^{d' \times d}$ is the weight matrix and $a \in \mathbb{R}^{2d'}$ is the attention vector. The output feature is $\boldsymbol{h}_i' = \sigma(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \boldsymbol{\Theta}\boldsymbol{h}_j)$. For more expressivity, multi-head attention is used to generate concatenated features of the form

$$\boldsymbol{h}_i' = \parallel_{k=1}^K \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \boldsymbol{\Theta}_k \boldsymbol{h}_j\right)$$

where $K$ is the number of attention heads.

## 3. SCATTERING ATTENTION LAYER

Inspired by the recent work in Sec. 2, we introduce an attention framework to combine multiple channels corresponding to GCN and scattering filters while adaptively assigning different weights to them based on filtered node features. While our full network used multi-head attention, we focus here on the processing performed independently by each attention head, deferring the multi-head configuration details to the general discussion of network architecture in the next section.

For every attention head, we first linearly transform the feature matrix $\boldsymbol{H}^{\ell-1}$ with a matrix $\boldsymbol{\Theta}$, setting the transformed feature matrix to be $\bar{\boldsymbol{H}}^{\ell} = \boldsymbol{H}^{\ell-1}\boldsymbol{\Theta}$. Then, based on the scattering GCN approach (Sec. 2.2 and [18]), a set of $C_{gcn}$ GCN channels and $C_{sct}$ scattering channels are calculated,

$$\begin{cases} \left.\begin{array}{l} \bar{\boldsymbol{H}}^{\ell}_{gcn,1} = \boldsymbol{A}\bar{\boldsymbol{H}}^{\ell} \\ \vdots \\ \bar{\boldsymbol{H}}^{\ell}_{gcn,C_{sct}} = \boldsymbol{A}^{C_{sct}}\bar{\boldsymbol{H}}^{\ell} \end{array}\right\} \text{GCN channels,} \\ \left.\begin{array}{l} \bar{\boldsymbol{H}}^{\ell}_{sct,1} = |\boldsymbol{U}_{p_1}\bar{\boldsymbol{H}}^{\ell}|^q \\ \vdots \\ \bar{\boldsymbol{H}}^{\ell}_{sct,C_{sct}} = |\boldsymbol{U}_{p_{C_{sct}}}\bar{\boldsymbol{H}}^{\ell}|^q \end{array}\right\} \text{scattering channels.} \end{cases} \tag{4}$$

The channels $\bar{\boldsymbol{H}}^{\ell}_{gcn,i}$ perform low-pass operations with different spatial support, aggregating information from 1,...,$C_{gcn}$-step neighborhoods, respectively, while $\bar{\boldsymbol{H}}^{\ell}_{sct,k}$, defined according to Eq. 3, enables band-pass filtering of graph signals.

Next, we compute attention coefficients that will be used in a shared attention layer to combine together the filtered channels. In order to compute these node-wise attention coefficient for each channel, we first compute

$$\boldsymbol{e}^{\ell}_{gcn,i} = \left[\bar{\boldsymbol{H}}^{\ell}\|\bar{\boldsymbol{H}}^{\ell}_{gcn,i}\right]\boldsymbol{a}, \quad \boldsymbol{e}^{\ell}_{sct,i} = \left[\bar{\boldsymbol{H}}_{\ell}\|\bar{\boldsymbol{H}}^{\ell}_{sct,i}\right]\boldsymbol{a},$$

with $\boldsymbol{a} \in \mathbb{R}^{2d_\ell}$ being a shared attention vector across all nodes and channels. Then, $\boldsymbol{e}^{\ell}_{gcn,j}, \boldsymbol{e}^{\ell}_{sct,k} \in \mathbb{R}^n$, can be considered as score vectors that indicate the importance of each channel.

Finally, the attention scores are normalized across all channels using the softmax function, yielding

$$\boldsymbol{\alpha}^{\ell}_{gcn,i} = \frac{\exp(\boldsymbol{e}^{\ell}_{gnc,i})}{\sum_{j=1}^{C_{gcn}}\exp(\boldsymbol{e}^{\ell}_{gcn,j}) + \sum_{k=1}^{C_{sct}}\exp(\boldsymbol{e}^{\ell}_{sct,k})},$$

with analogous $\boldsymbol{\alpha}^{\ell}_{sct,i}$. To obtain comparable weights when aggregating the $C_{gcn} + C_{sct} =: C$ channels, we set

$$\boldsymbol{H}^{\ell} = C^{-1}\sigma\left(\sum_{j=1}^{C_{gcn}}\boldsymbol{\alpha}^{\ell}_{gcn,j}\odot\bar{\boldsymbol{H}}^{\ell}_{gcn,j} + \sum_{k=1}^{C_{sct}}\boldsymbol{\alpha}^{\ell}_{sct,k}\odot\bar{\boldsymbol{H}}^{\ell}_{sct,k}\right),$$

where $\sigma(.) = \mathrm{LeakyReLU}(.)$ is used as nonlinearity here.

## 4. SCATTERING ATTENTION NETWORK

The full network architecture in all examples shown here uses one scattering attention layer (as described in Sec. 3), applied to the input node features, followed by a residual convolution layer (see Sec. 2.3), which then produces the output via a fully connected layer. We note that deeper configurations are possible, especially when processing big graphs, but for simplicity, we focus on having a single architecture for the network. This is similar to the design choice utilized in GAT [3]. Figure 1 illustrates this network structure and further technical details on each of its components are provided below.

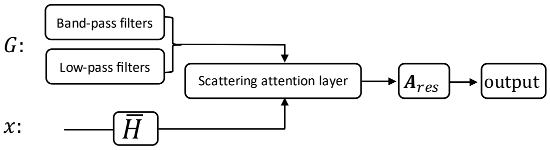

**Fig. 1**. Illustration of the proposed network architecture.

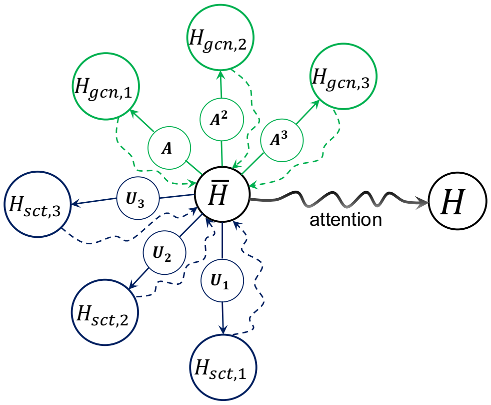

**Fig. 2**. Illustration of the proposed scattering attention layer. Attention weights are computed from a concatenation of the transformed layer input $\bar{\boldsymbol{H}}$ together with filtered signals that are first computed from it and then used to produce the layer output $\boldsymbol{H}$ via an attention-weighted linear combination.

**Attention layer configuration.** In this work, for simplicity, we set $C_{gcn} = C_{sct} = 3$, thus the attention layer combines three low-pass channels and three band-pass channels. The aggregation process of the attention layer is shown in Fig. 2 where $\boldsymbol{U}_{1,2,3}$ represents three first-order scattering transformations where $\boldsymbol{U}_1\boldsymbol{x} := \boldsymbol{\Psi}_1 x$, $\boldsymbol{U}_2\boldsymbol{x} := \boldsymbol{\Psi}_2 x$ and $\boldsymbol{U}_3\boldsymbol{x} := \boldsymbol{\Psi}_3 x$.

**Multihead attention.** Similar to other applications of attention mechanisms [3], we use multi-head attention here for stabilizing the training, thus rewriting the output (by a slight abuse of notation) of the $\ell$-th as

$$\boldsymbol{H}^{\ell} \longleftarrow \|_{\gamma=1}^{\Gamma}\boldsymbol{H}^{\ell}\left[\Theta \mapsto \Theta_{(\gamma)}; \alpha \mapsto \alpha_{(\gamma)}\right], \tag{5}$$

combining $\Gamma$ attention heads, where $\Gamma$ is tuned as a hyperparameter of the network.

**Residual convolution.** To eliminate the high frequency noise induced by the unlabeled nodes, the graph residual convolution (Sec. 2.3) is applied on the output of the concatenated multihead scattering scattering attention layer (Eq. 5), with $\alpha$ tuned as a hyperparameter of the network.

**Hyperparameter tuning.** All the datasets are split into train, validation and test sets. The validation set is used for searching hyperparameters and the test accuracy is shown in the

**Table 1**. Summary of dataset characteristics and node classification accuracy of previous methods compared to ours. Datasets are ordered by ascending homophily.

| Dataset | Classes | Nodes | Edges | Homophily | GCN | GAT | Sc-GCN | GSAN (ours) |
|---|---|---|---|---|---|---|---|---|
| Texas | 5 | 183 | 295 | 0.11 | 59.5 | 58.4 | 60.3 | 60.5 |
| Chameleon | 5 | 2,277 | 31,421 | 0.23 | 28.2 | 42.9 | 51.2 | 61.2 |
| CoraFull | 70 | 19,793 | 63,421 | 0.57 | 62.2 | 51.9 | 62.5 | 64.3 |
| Wiki-CS | 10 | 11,701 | 216,123 | 0.65 | 77.2 | 77.7 | 78.1 | 78.6 |
| Citeseer | 6 | 3,327 | 4,676 | 0.74 | 70.3 | 72.5 | 71.7 | 71.3 |
| Pubmed | 3 | 19,717 | 44,327 | 0.80 | 79.0 | 79.0 | 79.4 | 79.8 |
| Cora | 7 | 2,708 | 5,276 | 0.81 | 81.5 | 83.0 | 84.2 | 84.0 |
| DBLP | 4 | 17,716 | 52,867 | 0.83 | 59.3 | 66.1 | 81.5 | 82.6 |

presented results in Sec. 5. Tuned hyperparameters include the number of heads $\Gamma$, the residual parameter $\alpha$ and the width (i.e., number of neurons) of the channels. The tuning was performed using grid search guided by validation accuracy.

## 5. RESULTS

To evaluate our geometric scattering attention network (GSAN), we apply it to semi-supervised node classification and compare its results on several benchmarks to two popular graph neural networks (namely GCN [2] and GAT [3]), as well as the original Sc- GCN, which does not utilize attention mechanisms and is instead tuned via extensive hyperparameter grid search. These methods were applied to eight benchmark datasets of varied sizes and homophily (i.e., average class similarity across edges), as shown in Table 1. Texas and Chameleon are low-homophily datasets where nodes correspond to webpages and edges to links between them, with classes corresponding to webpage topic or monthly traffic (discretized into five levels), respectively [22]. Wiki-CS is a recently proposed benchmark, where the nodes represent computer science articles and the edges represent the hyperlinks [23]. The rest of the datasets are citation networks from different sources (i.e., Cora, Citeseer, Pubmed, DBLP), where nodes correspond to papers edges to citations [24, 25]. CoraFull is the larger version of the Cora dataset [26].

The results in Table 1 indicate that we improve upon previous methods, including Sc-GCN, which also uses a hybrid approach, but requires more intricate hyperparameter tuning to balance low-pass and band-pass channels [18]. The attention weights $\boldsymbol{\alpha}$, which are computed separately for every node (see Sec. 3), can also help understand the utilization of different channels in different regions of the graph. To demonstrate such analysis, we compute for each node the ratio between the attention it pays to band-pass channels compared to low-pass ones. We sum up the total attention $\sum_{i=1}^{C_{sct}} \mathbf{1}_n^T \boldsymbol{\alpha}_{sct,i}$ in the three scattering channels $\boldsymbol{U}_{1,2,3}$ and $\sum_{i=1}^{C_{gcn}} \mathbf{1}_n^T \boldsymbol{\alpha}_{gcn,i}$ in the three GCN channels $\boldsymbol{A}^{1,2,3}$ over nodes and heads. Finally, we compute the ratio $\frac{\sum_{i=1}^{C_{sct}} \mathbf{1}_n^T \boldsymbol{\alpha}_{sct,i}}{\sum_{i=1}^{C_{gcn}} \mathbf{1}_n^T \boldsymbol{\alpha}_{gcn,i}}$ between band-pass and low-pass channels. Figure 3 shows the distribution of these attention scores for four of the benchmark datasets, providing additional insight into the information utilized by our network.

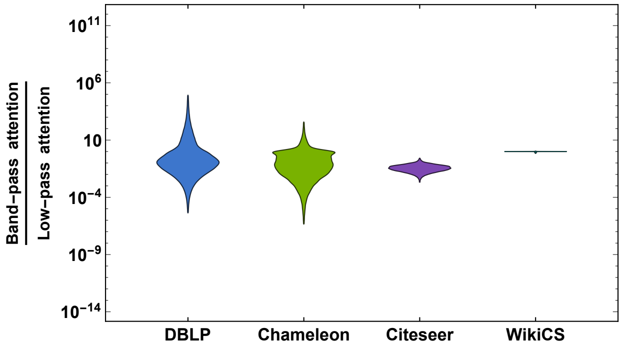

**Fig. 3**. Distribution of attention ratio between band-pass (scattering) and low-pass (GCN) channels across nodes and heads in DBLP, Chameleon, Citeseer, and WikiCS. Wider spread, indicating highly varied channel utilization, is exhibited by DBLP and Chameleon where the hybrid approach of GSAN (and Sc-GCN) achieves significant improvement over GCN and GAT. While Citeseer and Wiki-CS have similar spread, the latter exhibits higher attention to band-pass channels, which can be attributed to lower homophily.

## 6. CONCLUSIONS

The presented geometric scattering attention network (GSAN) introduces a new approach that leverages node-wise attention to incorporate together geometric scattering [11–14] and GCN [2] channels to form a hybrid model, further advancing the recently proposed Sc-GCN from [18]. Beyond its efficacy in semi-supervised node classification, the distribution of its learned attention scores provides a promising tool to study the spectral composition of information extracted from node features. We expect this to enable future work to distill tractable notions of regularity on graph to better understand how to leverage their intrinsic structure in geometric deep learning.

## 7. REFERENCES

[1] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[2] Thomas N Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[3] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018.

[4] Will Hamilton, Zhitao Ying, and Jure Leskovec, "Inductive representation learning on large graphs," in *NeurIPS*, 2017, pp. 1024–1034.

[5] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[6] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *NIPS*, 2016.

[7] Hoang NT and Takanori Maehara, "Revisiting graph neural networks: All we have is low-pass filters," *arXiv preprint arXiv:1905.09550*, 2019.

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

[9] Qimai Li, Zhichao Han, and Xiao-Ming Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 2018.

[10] Kenta Oono and Taiji Suzuki, "Graph neural networks exponentially lose expressive power for node classification," in *International Conference on Learning Representations*, 2019.

[11] Feng Gao, Guy Wolf, and Matthew Hirn, "Geometric scattering for graph data analysis," in *International Conference on Machine Learning*, 2019, pp. 2122–2131.

[12] Fernando Gama, Alejandro Ribeiro, and Joan Bruna, "Diffusion scattering transforms on graphs," in *International Conference on Learning Representations*, 2019.

[13] Fernando Gama, Alejandro Ribeiro, and Joan Bruna, "Stability of graph scattering transforms," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, vol. 32, pp. 8038–8048.

[14] Dongmian Zou and Gilad Lerman, "Graph convolutional neural networks via scattering," *Applied and Computational Harmonic Analysis*, vol. 49, no. 3, pp. 1046–1074, 2020.

[15] Stéphane Mallat, "Group invariant scattering," *Communications on Pure and Applied Mathematics*, vol. 65, no. 10, pp. 1331–1398, 2012.

[16] Joan Bruna and Stéphane Mallat, "Invariant scattering convolution networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1872–1886, August 2013.

[17] Joakim Andén and Stéphane Mallat, "Deep scattering spectrum," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4114–4128, August 2014.

[18] Yimeng Min, Frederik Wenkel, and Guy Wolf, "Scattering gcn: Overcoming oversmoothness in graph convolutional networks," *arXiv preprint arXiv:2003.08414*, 2020.

[19] Laurent Sifre and Stéphane Mallat, "Rotation, scaling and deformation invariant scattering for texture discrimination," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[20] Vincent Lostanlen and Stéphane Mallat, "Wavelet scattering on the pitch spiral," in *Proceedings of the 18th International Conference on Digital Audio Effects*, 2015, pp. 429–432.

[21] R.R. Coifman and M. Maggioni, "Diffusion wavelets," *ACHA*, vol. 21, no. 1, pp. 53–94, 2006.

[22] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang, "Geom-gcn: Geometric graph convolutional networks," in *International Conference on Learning Representations*, 2019.

[23] Péter Mernyei and Cătălina Cangea, "Wiki-cs: A wikipedia-based benchmark for graph neural networks," *arXiv preprint arXiv:2007.02901*, 2020.

[24] Zhilin Yang, William Cohen, and Ruslan Salakhudinov, "Revisiting semi-supervised learning with graph embeddings," in *Proceedings of The 33rd International Conference on Machine Learning*, 2016, vol. 48 of *PMLR*, pp. 40–48.

[25] Shirui Pan, Jia Wu, Xingquan Zhu, Chengqi Zhang, and Yang Wang, "Tri-party deep network representation," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 2016, pp. 1895–1901.

[26] Aleksandar Bojchevski and Stephan Günnemann, "Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking," in *International Conference on Learning Representations*, 2018.