# Power up! Robust Graph Convolutional Network via Graph Powering

**Ming Jin**[*] [2], **Heng Chang**[*†] [1], **Wenwu Zhu** [3], **Somayeh Sojoudi** [1,4]
[1] Tsinghua-Berkeley Shenzhen Institute, Tsinghua University
[2] Department of Electrical and Computer Engineering, Virginia Tech
[3] Department of Computer Science and Technology, Tsinghua University
[4] Department of Electrical Engineering and Computer Sciences, University of California at Berkeley
jinming@vt.edu, changh17@mails.tsinghua.edu.cn, wwzhu@tsinghua.edu.cn, sojoudi@berkeley.edu

## Abstract

Graph convolutional networks (GCNs) are powerful tools for graph-structured data. However, they have been recently shown to be vulnerable to topological attacks. To enhance adversarial robustness, we go beyond spectral graph theory to robust graph theory. By challenging the classical graph Laplacian, we propose a new convolution operator that is provably robust in the spectral domain and is incorporated in the GCN architecture to improve expressivity and interpretability. By extending the original graph to a sequence of graphs, we also propose a robust training paradigm that encourages transferability across graphs that span a range of spatial and spectral characteristics. The proposed approaches are demonstrated in extensive experiments to simultaneously improve performance in both benign and adversarial situations.

## Introduction

Graph convolutional networks (GCNs) are powerful extensions of convolutional neural networks (CNN) to graph-structured data. Recently, GCNs and variants have been applied to a wide range of domains, achieving state-of-the-art performances in social networks (Kipf and Welling 2017), traffic prediction (Rahimi, Cohn, and Baldwin 2018), recommendation systems (Ying et al. 2018), applied chemistry and biology (Kearnes et al. 2016; Fout et al. 2017), and natural language processing (Atwood and Towsley 2016; Hamilton, Ying, and Leskovec 2017; Bastings et al. 2017; Marcheggiani and Titov 2017), just to name a few (Zhou et al. 2018; Wu et al. 2019).

GCNs belong to a family of *spectral methods* that deal with spectral representations of graphs (Zhou et al. 2018; Wu et al. 2019). A fundamental ingredient of GCNs is the graph convolution operation defined by the graph Laplacian in the Fourier domain:

$$\boldsymbol{g_\theta} \star \boldsymbol{x} \coloneqq \hat{\boldsymbol{g}}_{\boldsymbol{\theta}}(\boldsymbol{L})\boldsymbol{x}, \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the graph signal on the set of vertices $\mathcal{V}$ and $\hat{\boldsymbol{g}}_{\boldsymbol{\theta}}$ is a spectral function applied to the graph Laplacian $\boldsymbol{L} \coloneqq \boldsymbol{D} - \boldsymbol{A}$ (where $\boldsymbol{D}$ and $\boldsymbol{A}$ are the degree matrix and the adjacency matrix, respectively). Because this operation is computational intensive for large graphs and non-spatially localized (Bruna et al. 2014), early attempts relied on a parameterization with smooth coefficients (Henaff, Bruna, and LeCun 2015) or a truncated expansion in terms of of Chebyshev polynomials (Hammond, Vandergheynst, and Gribonval 2011). By further restricting the Chebyshev polynomial order by 2, the approach in (Kipf and Welling 2017) referred henceforth as the vanilla GCN pushed the state-of-the-art performance of semi-supervised learning. The network has the following layer-wise update rule:

$$\boldsymbol{H}^{(l+1)} \coloneqq \psi\left(\mathcal{A}\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}\right), \tag{2}$$

where $\boldsymbol{H}^{(l)}$ is the $l$-th layer hidden state (with $\boldsymbol{H}^{(1)} \coloneqq \boldsymbol{X}$ as nodal features), $\boldsymbol{W}^{(l)}$ is the $l$-th layer weight matrix, $\psi$ is the usual point-wise activation function, and $\mathcal{A}$ is the convolution operator chosen to be the degree weighted Laplacian with some slight modifications (Kipf and Welling 2017). Subsequent GCN variants have different architectures, but they all share the use of the Laplacian matrix as the convolution operator (Zhou et al. 2018; Wu et al. 2019).

### Why Not Graph Laplacian?

Undoubtedly, the Laplacian operator (and its variants, e.g., normalized/powered Laplacian) plays a central role in spectral theory, and is a natural choice for a variety of spectral algorithms such as principal component analysis, clustering and linear embeddings (Chung and Graham 1997; Belkin and Niyogi 2002). *So what can be problematic?*

From a spatial perspective, GCNs with $d$ layers cannot acquire nodal information beyond its $d$-distance neighbors; hence, it severely limits its scope of data fusion. Recent works (Lee et al. 2018; Abu-El-Haija et al. 2018, 2019; Wu et al. 2019) alleviated this issue by directly powering the graph Laplacian.

From a spectral perspective, one could demand better *spectral properties*, given that GCN is fundamentally a particular (yet effective) approximation of the spectral convolution (1). A key desirable property for generic spectral methods is known as "spectral separation," namely the spectrum should comprise a few dominant eigenvalues whose associated eigenvectors reveal the sought structure in the graph. A well-known prototype is the Ramanujan property, for which the second

[*] The two first authors made equal contributions.
[†] This work was conducted during Heng Chang's visit to Professor Somayeh Sojoudi's group at UC Berkeley.
Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

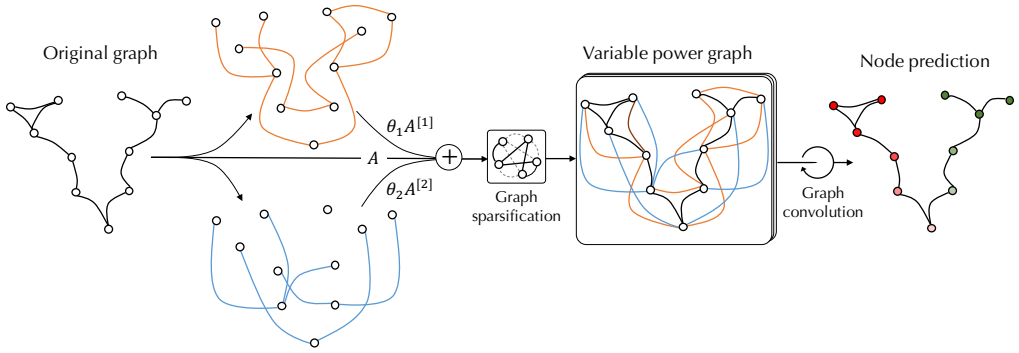


Figure 1: From the original graph, we generate a series of graphs, which are weighted by parameters of influence strengths, sparsified, and eventually combined to form a variable power graph.

leading eigenvalue of a $r$-regular graph is no larger than $2\sqrt{r-1}$, which is also enjoyed asymptotically by random $r$-regular graphs (Friedman 2004) and Erdős-Rényi graphs that are not too sparse (Feige and Ofek 2005). In a more realistic scenario, consider the stochastic block model (SBM), which attempts to capture the essence of many networks, including social graphs, citation graphs, and even brain networks (Holland, Laskey, and Leinhardt 1983).

**Definition 1** (Simplified stochastic block model)**.** The graph $\mathcal{G}$ with $n$ nodes is drawn under SBM$(n, k, a_{\text{intra}}, a_{\text{inter}})$ if the nodes are evenly and randomly partitioned into $k$ communities, and nodes $i$ and $j$ are connected with probability $a_{\text{intra}}/n \in [0, 1]$ if they belong to the same community, and $a_{\text{inter}}/n \in [0, 1]$ if they are from different communities.

It turns out that for community detection, the top $k$ leading eigenvectors of the adjacency matrix $\boldsymbol{A}$ play an important role. In particular, for the case of 2 communities, spectral bisection algorithms simply take the second eigenvector to reveal the community structure. This can be also seen from the expected adjacency matrix $\mathbb{E}[\boldsymbol{A}]$ under SBM$(n, 2, a_{\text{intra}}, a_{\text{inter}})$, which is a rank-2 matrix with the top eigenvalue $\frac{1}{2}(a_{\text{intra}} + a_{\text{inter}})$ and eigenvector $\mathbf{1}$, and the second eigenvalue $\frac{1}{2}(a_{\text{intra}} - a_{\text{inter}})$ and eigenvector $\boldsymbol{\sigma}$ such that $\sigma_i = 1$ if $i$ is in community 1 and $\sigma_i = -1$ otherwise. More generally, the second eigenvalue is of particular theoretical interests because it controls at the first order how fast heat diffuses through graph, as depicted by the discrete Cheeger inequality (Lee, Gharan, and Trevisan 2014).

While one would expect taking the second eigenvector of the adjacency matrix suffices, it often fails in practice (even when it is theoretically possible to recover the clusters given the signal-to-noise ratio). This is especially true for sparse networks, whose average nodal degrees is a constant that does not grow with the network size. This is because the spectrum of the Laplacian or adjacency matrix is blurred by "outlier" eigenvalues in the sparse regime, which is often caused by high degree nodes (Kaufmann, Bonald, and Lelarge 2016). Unsurprisingly, powering the Laplacian would be of no avail, because it does not change the eigenvectors or the ordering of eigenvalues. In fact, those outliers can become more salient after powering, thereby weakening the useful spectral signal even further. Besides, pruning the largest degree nodes in the adjacency matrix or normalizing the Laplacian cannot solve the issue. To date, the best results for pruning does not apply down to the theoretical recovery threshold (Coja-Oghlan 2010; Mossel, Neeman, and Sly 2012; Le, Levina, and Vershynin 2015); either outliers would persist or one could prune too much that the graph is destroyed. As for normalized Laplacian, it may overcorrect the large degree nodes, such that the leading eigenvectors would catch the "tails" of the graph, i.e., components weakly connected to the main graph. See the Appendix for an experimental illustration.

**In summary, graph Laplacian may not be the ideal choice due to its limited spatial scope of information fusion, and its undesirable artefacts in the spectral domain.**

## If Not Laplacian, Then What?

In searching for alternatives, potential choices are many, so it is necessary to clarify the goals. In view of the aforementioned pitfalls of the graph Laplacian, one would naturally ask the question:

**Can we find an operator that has wider spatial scope, more robust spectral properties, and is meanwhile interpretable and can increase the expressive power of GCNs?**

From a perspective of *graph data analytics*, this question gauges how information is propagated and fused on a graph, and how we should interpret "adjacency" in a much broader sense. An image can be viewed as a regular grid, yet the operation of a CNN filter goes beyond the nearest pixel to a local neighborhood to extract useful features. How to draw an analogy to graphs?

From a perspective of *robust learning*, this question sheds light on the basic observation that real-world graphs are often noisy and even adversarial. The nice spectral properties of a graph topology can be lost with the presence or absence of edges. What are some principled ways to robustify the convolution operator and graph embeddings?

In this paper, we propose a graph learning paradigm that aims at achieving this goal, as illustrated in Figure 1. The key idea is *to generate a sequence of graphs from the given graph that capture a wide range of spectral and spatial behaviors.* We propose a new operator based on this derived sequence.

**Definition 2** (Variable power operator)**.** Consider an unweighted and undirected graph $\mathcal{G}$. Let $\boldsymbol{A}^{[k]}$ denote the $k$-

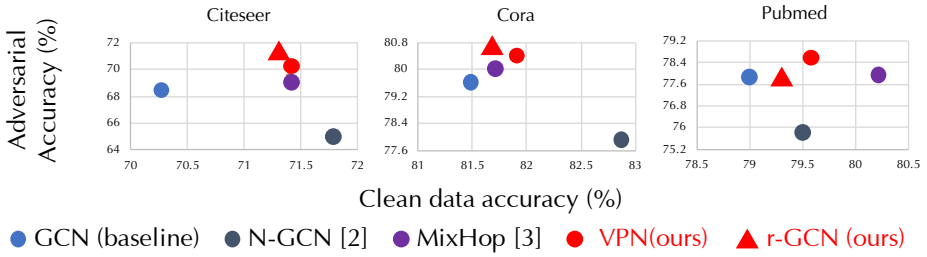


Figure 2: Our proposed framework can improve both clean and adversarial (10% attack by $\mathcal{A}_{DW_3}$ (Bojchevski and Günnemann 2019)) accuracy for semi-supervised learning benchmarks.

distance adjacency matrix, i.e., $\left[\boldsymbol{A}^{[k]}\right]_{ij} = 1$ if and only if the shortest distance (in the original graph) between nodes $i$ and $j$ is $k$. The variable power operator of order $r$ is defined as:

$$\boldsymbol{A}_{\boldsymbol{\theta}}^{(r)} = \sum_{k=0}^{r} \theta_k \boldsymbol{A}^{[k]}, \tag{3}$$

where $\boldsymbol{\theta} \coloneqq (\theta_0, \ldots, \theta_r)$ is a set of parameters.

Clearly, $\boldsymbol{A}_{\boldsymbol{\theta}}^{(r)}$ is a natural extension of the classical adjacency matrix (i.e., $r = 1$ and $\theta_0 = \theta_1 = 1$). With power order $r > 1$, one can increase the spatial scope of information fusion on the graph when applying the convolution operation. The parameters $\theta_k$ also has a natural explanation—the magnitude and the sign of $\theta_k$ can be viewed as "global influence strength" and "global influence propensity" at distance $k$, respectively, which also determines the participation factor of each graph in the sequence in the aggregated operator.

Furthermore, we provide some theoretical justification of the proposed operator by establishing the following asymptotic property of spectral separation under the important SBM setting, which is, nevertheless, not enjoyed by the classical Laplacian operator or its normalized or powered versions. (All proofs are given in the Appendix.)

**Theorem 3** (Asymptotic spectral separation of variable power operator)**.** Consider a graph $\mathcal{G}$ drawn from $\text{SBM}(n, 2, a_{\text{intra}}, a_{\text{inter}})$. Assume that the signal-to-noise ratio $\xi_2^2/\xi_1 > 1$, where $\xi_1 = \frac{1}{2}(a_{\text{intra}} + a_{\text{inter}})$ and $\xi_2 = \frac{1}{2}(a_{\text{intra}} - a_{\text{inter}})$ (c.f., (Decelle et al. 2011)). Suppose $r$ is on the order of $c\log(n)$ for a constant $c$, such that $c\log(\xi_1) < 1/4$. Given nonvanishing $\theta_k$ for $k > r/2$, the variable power operator $\boldsymbol{A}_{\boldsymbol{\theta}}^{(r)}$ has the following spectral properties: **(i)** the leading eigenvalue is on the order of $\Theta\left(\|\boldsymbol{\theta}\|_1 \xi_1^r\right)$, the second leading eigenvalue is on the order of $\Theta\left(\|\boldsymbol{\theta}\|_1 \xi_2^r\right)$, and the rest are bounded by $\|\boldsymbol{\theta}\|_1 n^{\epsilon} \xi_1^{r/2} O(\log(n))$ for any fixed $\epsilon > 0$; and **(ii)** the two leading eigenvectors are sufficient to recover the two communities asymptotically (i.e., as $n$ goes to infinity).

Theorem 3 is the weighted analogue of Theorem 2 from (Abbe et al. 2018). Intuitively, the above theoretical result suggests that the variable power operator is able to *"magnify" benign signals* from the latent graph structure while *"suppressing" noises* due to random artefacts. This is expected to improve spectral methods in general, especially when the benign signals tend to be overwhelmed by noises. For the rest of the paper, we will apply this insight to propose a robust graph learning paradigm in Section , as well as a new GCN architecture in Section . We also provide empirical evidence of the gain from this theory in Section  and conclude in Section .

## Related Works

**Beyond nearest neighbors.** Several works have been proposed to address the issue of limited spatial scope by powering the adjacency matrix (Lee et al. 2018; Wu et al. 2019; Li et al. 2019). However, simply powering the adjacency does not extract spectral gap and may even make the eigenspectrum more sensitive to perturbations. (Abu-El-Haija et al. 2018, 2019) also introduced weight matrices for neighbors at different distances. But this could substantially increase the risk of overfitting in the low-data regime and make the network vulnerable to adversarial attacks.

**Robust spectral theory.** The robustness of spectral methods has been extensively studied for graph partitioning/clustering (Li et al. 2007; Balakrishnan et al. 2011; Chaudhuri, Chung, and Tsiatas 2012; Amini et al. 2013; Joseph, Yu et al. 2016; Diakonikolas et al. 2019). Most recently, operators based on self-avoiding or nonbacktracking walks have become popular for SBM (Massoulié 2014; Mossel, Neeman, and Sly 2013; Bordenave, Lelarge, and Massoulié 2015), which provably achieve the detection threshold conjectured by (Decelle et al. 2011). Our work is partly motivated by the graph powering approach by (Abbe et al. 2018), which leveraged the result of (Massoulié 2014; Bordenave, Lelarge, and Massoulié 2015) to prove the spectral gap. The main difference with this line of work is that these operators are studied only for spectral clustering without incorporating nodal features. Our proposed variable power operator can be viewed as a kernel version of the graph powering operator (Abbe et al. 2018), thus substantially increasing the capability of learning complex nodal interactions while maintaining the spectral property.

**Robust graph neural network.** While there is a surge of adversarial attacks on graph neural networks (GNNs) (Dai et al. 2018; Zügner and Günnemann 2019; Bojchevski and Günnemann 2019), very few methods have been proposed for defense (Sun et al. 2018). Existing works employed known techniques from computer vision (Szegedy et al. 2014; Goodfellow, Shlens, and Szegedy 2015; Szegedy et al. 2016), such

as adversarial training with "soft labels" (Chen et al. 2019) or outlier detection in the hidden layers (Zhu et al. 2019), but they do not exploit the unique characteristics of graph-structured data. Importantly, our approach simultaneously improves performance in both the benign and adversarial tests, as shown in Figure 2 (details are presented in Section ).

## Graph Augmentation: Robust Training Paradigm

**Exploration of the spectrum of spectral and spatial behaviors.** Given a graph $\mathcal{G}$, consider a family of its powered graphs, $\{\mathcal{G}^{(2)}, \ldots, \mathcal{G}^{(r)}\}$, where $\mathcal{G}^{(k)}$ is obtained by connecting nodes with distance less than or equal to $k$. This "graph augmentation" procedure is similar to "data augmentation", because instead of limiting the learning on a single graph that is given, we artificially create a series of graphs that are closely related to each other in the spatial and spectral domains.

As we increase the power order, the graph also becomes more homogenized. In particular, it can help near-isolated nodes (i.e., low-degree vertices), since they become connected beyond their nearest neighbors. By comparison, simply raising the adjacency matrix to its $r$-th power will make them appear even more isolated, because it inadvertently promotes nodes with high degrees or nearby loops much more substantially as a result of feedback. Furthermore, the powered graphs can extract spectral gaps in the original graph despite local irregularities, thus boosting the signal-to-noise ratio in the spectral domain.

**Transfer of knowledge from the powered graph sequence.** Consider a generic learning task on a graph $\mathcal{G}$ with data $\mathcal{D}$. The loss function is denoted by $\ell(\mathcal{W}; \mathcal{G}, \mathcal{D})$ for a particular GCN architecture parametrized by $\mathcal{W}$. For instance, in semi-supervised learning, $\mathcal{D}$ consists of features and labels on a small set of nodes, and $\ell$ is the cross-entropy loss over all labeled examples. Instead of minimizing over $\ell(\mathcal{W}; \mathcal{G}, \mathcal{D})$ alone, we use all the powered graphs:

$$\ell(\mathcal{W}; \mathcal{G}, \mathcal{D}) + \sum_{k=2}^{r} \alpha_k \ell(\mathcal{W}; \mathcal{G}^{(k)}, \mathcal{D}), \qquad \text{(r-GCN)}$$

where $\alpha_k \geq 0$ gauges how much information one desires to transfer from powered graph $\mathcal{G}^{(k)}$ to the learning process. By minimizing the (r-GCN) objective, one seeks to optimize the network parameter $\mathcal{W}$ on multiple graphs simultaneously, which is beneficial in two ways: **(i)** in the *low-data regime*, like semi-supervised learning, it helps to reduce the variance to improve generalization and transferability; **(ii)** in the *adversarial setting*, it robustifies the network since it is more likely that the perturbed network is contained in the wider spectrum during training.

## From Fixed to Variable Power Network

By using the variable power operator illustrated in Figure 1, we substantially increase the search space of graph operators. The proposed operator also leads to a new family of graph algorithms with broader scope of spatial fusion and enhanced spectral robustness. As the power grows, the network eventually becomes dense. To manage this increased complexity and make the network more robust against adversarial attacks in the feature domain, we propose a pruning mechanism.

**Graph sparsification.** Given a graph $\mathcal{G} \coloneqq (\mathcal{V}, \mathcal{E}^{[1]})$, consider its powered version $\mathcal{G}^{(r)} \coloneqq (\mathcal{V}, \mathcal{E}^{(r)})$ and a sequence of intermediate graphs $\mathcal{G}^{[2]}, \ldots, \mathcal{G}^{[r]}$, where $\mathcal{G}^{[k]} \coloneqq (\mathcal{V}, \mathcal{E}^{[k]})$ is constructed by connecting two vertices if and only if the shortest distance is $k$ in $\mathcal{G}$. Clearly, $\{\mathcal{E}^{[k]}\}_{k=1}^{r}$ forms a partition of $\mathcal{E}^{(r)}$. For each node $i \in \mathcal{V}$, denote its $r$-neighborhood by $\mathcal{N}_r(i) \coloneqq \{j \in \mathcal{V} \mid d_{\mathcal{G}}(i,j) \leq r\}$, which is identical to the set of nodes adjacent to $i$ in $\mathcal{G}^{[r]}$. Next, for each edge within this neighborhood, we associate a value using some suitable distance metric $\phi$ to measure "aloofness." For instance, it can be the usual Euclidean distance or correlation distance in the feature space. Based on this formulation, we prune an edge $e \coloneqq (i,j)$ in $\mathcal{E}^{(r)}$ if the value is larger than a threshold $\tau(i,j)$, and denote the edge set after pruning $\bar{\mathcal{E}}^{(r)}$. Then, we can construct a new sequence of sparsified graphs, $\overline{\mathcal{G}}^{[k]}$ with edge sets $\bar{\mathcal{E}}^{[k]} = \mathcal{E}^{[k]} \cap \bar{\mathcal{E}}^{(r)}$ and adjacency matrix $\bar{\boldsymbol{A}}^{[k]}$. Hence, the variable power operator is given by $\bar{\boldsymbol{A}}_{\boldsymbol{\theta}}^{(r)} = \sum_{k=0}^{r} \theta_k \bar{\boldsymbol{A}}^{[k]}$. Due to the influence of high-degree nodes in the spectral domain, one can *adaptively* choose the thresholds $\tau(i,j)$ to mitigate their effects. Specifically, we choose $\tau$ to be a small number if either $i$ or $j$ are nodes with high degrees, thereby making the sparsification more influential in densely connected neighborhoods than weakly connected parts.

**Layer-wise update rule.** To demonstrate the effectiveness of the proposed operator, we adopt the vanilla GCN strategy (2). Importantly, we replace the graph convolutional operator $\mathcal{A}$ with the variable power operator to obtain the variable power network (VPN):

$$\mathcal{A} = \boldsymbol{D}^{-\frac{1}{2}}(\boldsymbol{I} + \bar{\boldsymbol{A}}_{\boldsymbol{\theta}}^{(r)})\boldsymbol{D}^{-\frac{1}{2}}, \qquad \text{(VPN)}$$

where $D_{ii} = 1 + |\{j \in \mathcal{V} \mid d_{\mathcal{G}}(i,j) = 1\}|$. The introduction of $\boldsymbol{I}$ is reminiscent of the "renormalization trick" (Kipf and Welling 2017), but it can be also viewed as a regularization strategy in this context, which is well-known to improve the spectral robustness (Amini et al. 2013; Joseph, Yu et al. 2016). This construction immediately increases the scope of data fusion by a factor of $r$.

**Proposition 4.** By choosing $\mathcal{A}$ with (VPN) in the layer-wise update rule (2), the output at each node from a $L$-layer GCN depends on neighbors within $L * r$ hops.

Since we proved that the variable power operator has nice spectral separation in Theorem 3, VPN is expected to promote useful spectral signals from the graph topology (similar to the preservation of useful information in images (Jacobsen, Smeulders, and Oyallon 2018), our method preserves useful information in the graph topology). This claim is substantiated with the following proposition.

**Proposition 5.** Given a graph with two hidden communities. Consider a 2-layer GCN architecture with layer-wise update rule (2). Suppose that $\mathcal{A}$ has a spectral gap. Further, assume that the leading two eigenvectors are asymptotically aligned

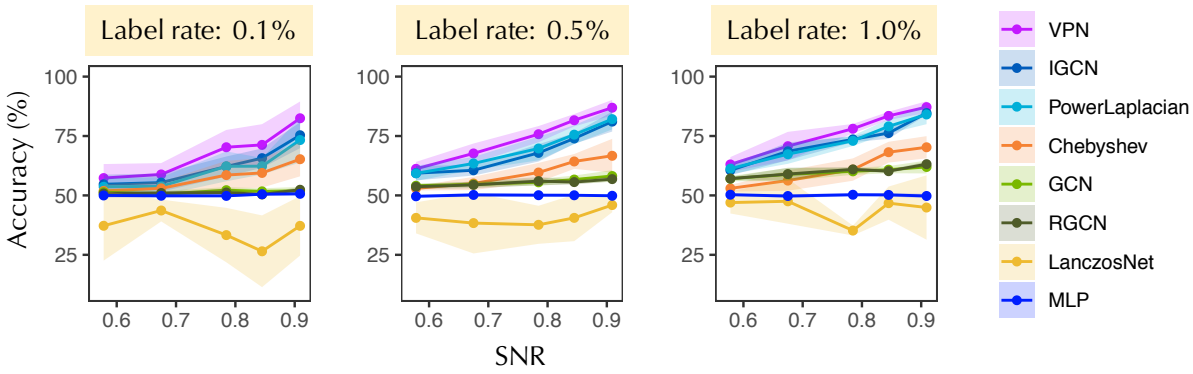


Figure 3: Comparison for SBM dataset. Shaded area indicates standard deviation over 10 runs.

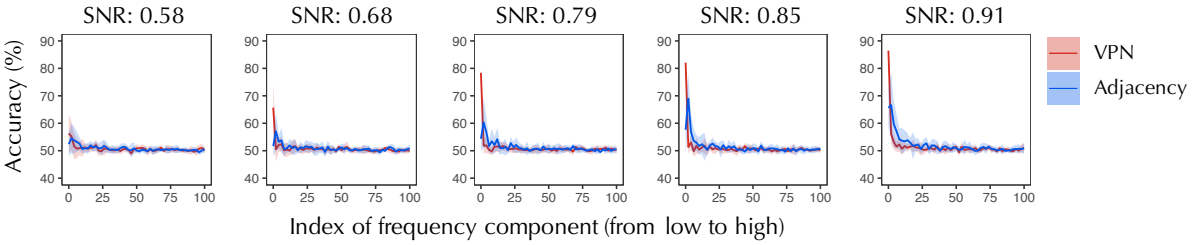


Figure 4: Accuracy of first 100 frequency components of VPN (red) and adjacency matrix (blue).

with $\mathbf{1}$ and $\boldsymbol{\nu}$, i.e., the community membership vector, and that both are in the range of feature matrix $\boldsymbol{X}$. Then, there exists a configuration of $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$ such that the GCN outputs can recover the community with high probability.

## Experiments

The proposed methods are evaluated against several recent models, including vanilla GCN (Kipf and Welling 2017) and its variant PowerLaplacian where we simply replace the adjacent matrix with its powered version, three baselines using powered Laplacian IGCN (Li et al. 2019), SGC (Wu et al. 2019) and LNet (Liao et al. 2019), the recent method MixHop (Abu-El-Haija et al. 2019) which attempts to increase spatial scope fusion, as well as a state-of-the-art baseline and RGCN (Zhu et al. 2019), which is also aimed at improving the robustness of Vanilla GCN. All baseline methods on based on their public codes.

### Revisiting Stochastic Block Model

**SBM dataset.** We generated a set of networks under SBM with 4000 nodes and parameters such that the SNRs range from 0.58 to 0.91. To disentangle the effects from nodal features with that from the spectral signal, we set the nodal features to be one-hot vectors. The label rates are 0.1%, 0.5% and 1%, the validation rate is 1%, and the rest of the nodes are testing points.

**Performance.** Since the nodal features do not contain any useful information, learning without topology such as multilayer perceptron (MLP) is only as good as random guessing. The incorporation of graph topology improves classification performance—the higher the SNR (i.e., $\xi_2^2/\xi_1$, see Theorem 3), the higher the accuracy. Overall, as shown in Figure 3, the performance of the proposed method (VPN) is superior than other baselines, which either use Laplacian (GCN, Chebyshev, RGCN, LNet) or its powered variants (IGCN, PowerLaplacian).

**Spectral separation and Fourier modes.** From the eigenspectrum of the convolution operators, we see that the spectral separation property is uniquely possessed by VPN, whose first two leading eigenvectors carry useful information about the underlying communities: without the help of nodal features, the accuracy is 87% even with label rate of 0.5%. Let $\boldsymbol{\Phi}$ denote the Fourier modes of the the adjacency matrix or VPN, and $\boldsymbol{X}$ be the nodal features (i.e., identity matrix). We analyze the information from spectral signals (e.g., the $k$-th and $k+1$-th eigenvectors) by estimating the accuracy of an MLP with filtered nodal features, namely $\boldsymbol{\Phi}_{:,k:(k+1)}\boldsymbol{\Phi}_{:,k:(k+1)}^{\top}\boldsymbol{X}$, as shown in Figure 4. The accuracy reflects the information content in the frequency components. We see that the two leading eigenvectors of VPN are sufficient to perform classification, whereas those of the adjacency matrix cannot make accurate inferences.

### Semi-supervised Node Classification

**Experimental setup.** We followed the setup of (Yang, Cohen, and Salakhutdinov 2016; Kipf and Welling 2017) for citation networks Citeseer, Cora and Pubmed (please refer to the Appendix for more details).

**Graph powering order** can influence spatial and spec-

| Model | Citeseer | Cora | Pubmed |
|---|---|---|---|
| ManiReg (Belkin, Niyogi, and Sindhwani 2006) | 60.1 | 59.5 | 70.7 |
| SemiEmb (Weston et al. 2012) | 59.6 | 59.0 | 71.1 |
| LP (Zhu, Ghahramani, and Lafferty 2003) | 45.3 | 68.0 | 63.0 |
| DeepWalk (Perozzi, Al-Rfou, and Skiena 2014) | 43.2 | 67.2 | 65.3 |
| ICA (Lu and Getoor 2003) | 69.1 | 75.1 | 73.9 |
| Planetoid (Yang, Cohen, and Salakhutdinov 2016) | 64.7 | 75.7 | 77.2 |
| Vanilla GCN (Kipf and Welling 2017) | 70.3 | 81.5 | 79.0 |
| PowerLaplacian | 70.5 | 80.5 | 78.3 |
| IGCN(RNM) (Li et al. 2019) | 69.0 | 80.9 | 77.3 |
| IGCN(AR) (Li et al. 2019) | 69.3 | 81.1 | 78.2 |
| LNet (Liao et al. 2019) | $66.2 \pm 1.9$ | $79.5 \pm 1.8$ | $78.3 \pm 0.3$ |
| RGCN (Zhu et al. 2019) | $71.2 \pm 0.5$ | $\mathbf{82.8} \pm 0.6$ | $79.1 \pm 0.3$ |
| SGC (Wu et al. 2019) | $\mathbf{71.9} \pm 0.1$ | $81.0 \pm 0.0$ | $78.9 \pm 0.0$ |
| MixHop (Abu-El-Haija et al. 2019) | $71.4 \pm 0.8$ | $81.9 \pm 0.4$ | $\mathbf{80.8} \pm 0.6$ |
| r-GCN (this paper) | $71.3 \pm 0.5$ | $81.7 \pm 0.2$ | $79.3 \pm 0.3$ |
| VPN (this paper) | $\mathbf{71.7} \pm 0.6$ | $\mathbf{82.3} \pm 0.3$ | $\mathbf{79.8} \pm 0.4$ |

Table 1: Results for semi-supervised node classification. We highlighted the best and the second best performances, where we broke the tie by choosing the one with the smallest standard deviation.

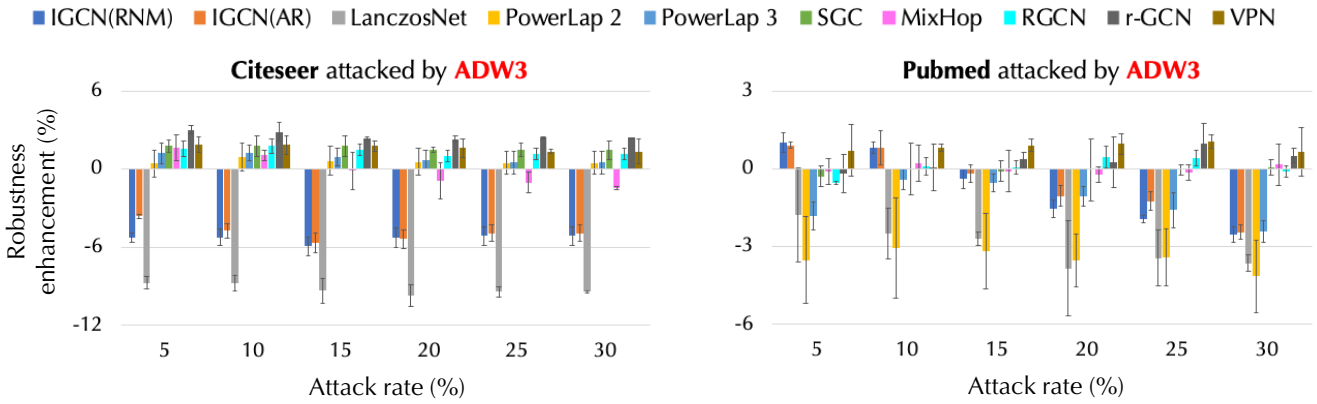


Figure 5: Robustness merit $\left(\text{accuracy}^{\text{post-attack}}_{\text{proposed-method}} - \text{accuracy}^{\text{post-attack}}_{\text{vanilla GCN}}\right)$ reported in percentage under $\mathcal{A}_{DW_3}$ attack. The error bar indicates standard deviation over 20 independent simulations.

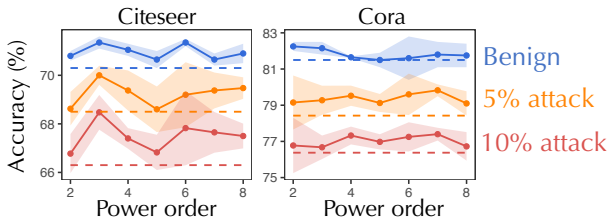


Figure 6: Benign and adversarial accuracy as power order increases. Dahsed lines correspond to vanilla GCN.

tral behaviors. Our theory suggests powering to the order of $\log(n)$; in practice, orders of 2 to 4 suffice (Figure 6). Here, we chose the power order to be 4 for r-GCN on Citeseer and Cora, and 3 for Pubmed, and reduced the order by 1 for VPN.

**Performance.** By replacing Laplacian with VPN, we see an immediate improvement in performance (Table 1). We also see that a succinct parametrization of the global influence relation in VPN is able to increase the expressivity of the network. For instance, the learned $\boldsymbol{\theta}$ at distances 2 and 3 for Citeseer are 3.15e-3 and 3.11e-3 with $p$-value less than 1e-5. This implies that the network tends to put more weights in closer neighbors.

## Defense Against Evasion Attacks

To evaluate the robustness of the learned network, we considered the setting of evasion attacks, where the model is trained on benign data but tested on adversarial data.

**Adversarial methods.** Five strong global attack methods are considered, including DICE (Zügner and Günnemann 2019), $\mathcal{A}_{abr}$ and $\mathcal{A}_{DW_3}$ (Bojchevski and Günnemann 2019), Meta-Train and Meta-Self (Zügner and Günnemann 2019). We further modulated the severity of attack methods by varying the attack rate, which corresponds to the percentage of

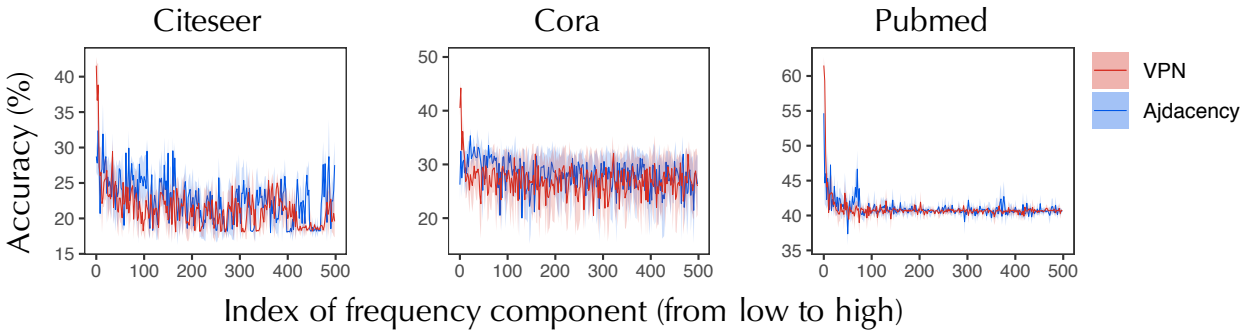


Figure 7: Accuracy of first 500 frequency components for VPN (red) and adjacency matrix (blue). The x axis corresponds to the index $k$ in $\boldsymbol{\Phi}_{:,k:(k+1)}\boldsymbol{\Phi}_{:,k:(k+1)}^{\top}\boldsymbol{X}$ for signal reconstruction.

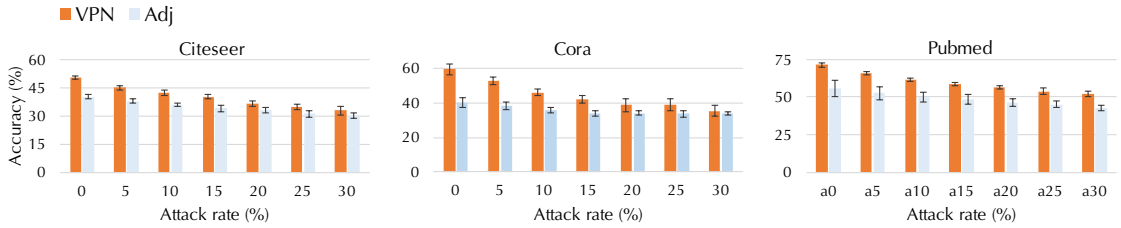


Figure 8: Accuracy of the leading frequency components in adversarial testing. We use $\widetilde{\boldsymbol{\Phi}}_{:,1:k}\widetilde{\boldsymbol{\Phi}}_{:,1:k}^{\top}\boldsymbol{X}$ as reconstructed nodal features for testing, where $k$ is 10 for Citeseer and 5 for Cora and Pubmed. Error bar indicates standard deviation over 10 runs.

edges to be attacked.

**Robustness evaluation.** In general, both r-GCN and VPN are able to improve over baselines for the defense against evasion attacks, e.g., Figure 5 for the $\mathcal{A}_{DW_3}$ attack (detailed results for other attacks are listed in the Appendix). It can be also observed that the proposed methods are more robust in Citeseer and Cora than Pubmed. In addition to the low label rates, we conjectured that topological attacks are more difficult to defend for networks with prevalent high-degree nodes, because the attacker can bring in more irrelevant vertices by simply adding a link to the high-degree nodes.

**Informative and robust low-frequency spectral signal.** It has been observed by (Wu et al. 2019; Maehara 2019) that GCNs share similar characteristics of low-pass filters, in the sense that nodal features filtered by low-frequency Fourier modes lead to accurate classifiers (e.g., MLP). However, one key question left unanswered is how to obtain the Fourier modes. In their experiments, they derive it from the graph Laplacian. By using VPN to construct the Fourier modes, we show that the information content in the low-frequency domain can be improved.

More specifically, we first perform eigendecomposition of the graph convolutional operator (i.e., graph Laplacian or VPN) to obtain the Fourier modes $\boldsymbol{\Phi}$. We then reconstruct the nodal features $\boldsymbol{X}$ using only the $k$-th and the $k+1$-th eigenvectors, i.e., $\boldsymbol{\Phi}_{:,k:(k+1)}\boldsymbol{\Phi}_{:,k:(k+1)}^{\top}\boldsymbol{X}$. We then use the reconstructed features in MLP to perform the classification task in a supervised learning setting. As Figure 7 shows, features filtered by the leading eigenvectors of VPN lead to higher classification accuracy compared to the classical adjacency matrix.

For the adversarial testing, we construct a new basis $\widetilde{\boldsymbol{\Phi}}$ based on the attacked graph, and then use $\widetilde{\boldsymbol{\Phi}}_{:,k:(k+1)}\widetilde{\boldsymbol{\Phi}}_{:,k:(k+1)}^{\top}\boldsymbol{X}$ as test points for the MLP trained in the clean data setting. As can be seen in Figure 8, models trained based on VPN filtered features also have better adversarial robustness in evasion attacks. Since the eigenvalues of the corresponding operator exhibit low-pass filtering characteristics, the enhanced benign and adversarial accuracy of VPN is attributed to the increased signal-to-noise ratio in the low-frequency domain. This is in alignment with the theoretical proof of spectral gap developed in this study.

## Conclusion

This study goes beyond classical spectral graph theory to defend GCNs against adversarial attacks. We challenge the central building block of existing methods, namely the graph Laplacian, which is not robust to noisy links. For adversarial robustness, spectral separation is a desirable property. We propose a new operator that enjoys this property and can be incorporated in GCNs to improve expressivity and interpretability. Furthermore, by generating a sequence of powered graphs based on the original graph, we can explore a spectrum of spectral and spatial behaviors and encourage transferability across graphs. The proposed methods are shown to improve both benign and adversarial accuracy over various baselines evaluated against a comprehensive set of attack strategies.

## Acknowledgements

This work is supported by the National Key Research and Development Program of China (No. 2020AAA0107800, 2018AAA0102000) National Natural Science Foundation of China Major Project (No. U1611461). Heng Chang is partially supported by the 2020 Tencent Rhino-Bird Elite Training Program.

## Ethics Statement

The main result of this work can benefit the family of graph convolutional networks, since the Laplacian operator is the fundamental building block of existing architectures. The method is able to improve accuracy and robustness to adversarial attacks simultaneously, which would further widen the applications of GCN in safety-critical applications, such as power grid and transportation.

## References

Abbe, E.; Boix, E.; Ralli, P.; and Sandon, C. 2018. Graph powering and spectral robustness. *arXiv preprint arXiv:1809.04818* .

Abu-El-Haija, S.; Kapoor, A.; Perozzi, B.; and Lee, J. 2018. N-GCN: Multi-scale graph convolution for semi-supervised node classification. *arXiv preprint arXiv:1802.08888* .

Abu-El-Haija, S.; Perozzi, B.; Kapoor, A.; Harutyunyan, H.; Alipourfard, N.; Lerman, K.; Steeg, G. V.; and Galstyan, A. 2019. MixHop: Higher-Order Graph Convolution Architectures via Sparsified Neighborhood Mixing. *arXiv preprint arXiv:1905.00067* .

Amini, A. A.; Chen, A.; Bickel, P. J.; Levina, E.; et al. 2013. Pseudo-likelihood methods for community detection in large sparse networks. *The Annals of Statistics* 41(4): 2097–2122.

Atwood, J.; and Towsley, D. 2016. Diffusion-convolutional neural networks. In *NIPS 2016*, 1993–2001.

Balakrishnan, S.; Xu, M.; Krishnamurthy, A.; and Singh, A. 2011. Noise thresholds for spectral clustering. In *NIPS 2011*, 954–962.

Bastings, J.; Titov, I.; Aziz, W.; Marcheggiani, D.; and Sima'an, K. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *EMNLP 2017*, 1957–1967.

Belkin, M.; and Niyogi, P. 2002. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS 2002*, 585–591.

Belkin, M.; Niyogi, P.; and Sindhwani, V. 2006. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. *Journal of Machine Learning Research* 7: 2399–2434.

Bojchevski, A.; and Günnemann, S. 2019. Adversarial Attacks on Node Embeddings via Graph Poisoning. In *International Conference on Machine Learning*, 695–704.

Bordenave, C.; Lelarge, M.; and Massoulié, L. 2015. Non-backtracking spectrum of random graphs: community detection and non-regular ramanujan graphs. In *FOCS 2015*, 1347–1357. IEEE.

Bruna, J.; Zaremba, W.; Szlam, A.; and LeCun, Y. 2014. Spectral networks and locally connected networks on graphs. In *ICLR 2014*.

Chaudhuri, K.; Chung, F.; and Tsiatas, A. 2012. Spectral Clustering of Graphs with General Degrees in the Extended Planted Partition Model. *Journal of Machine Learning Research* 2012: 1–23.

Chen, J.; Wu, Y.; Lin, X.; and Xuan, Q. 2019. Can Adversarial Network Attack be Defended? *arXiv preprint arXiv:1903.05994* .

Chung, F. R.; and Graham, F. C. 1997. *Spectral graph theory*. 92. American Mathematical Soc.

Coja-Oghlan, A. 2010. Graph partitioning via adaptive spectral techniques. *Combinatorics, Probability and Computing* 19(2): 227–284.

Dai, H.; Li, H.; Tian, T.; Huang, X.; Wang, L.; Zhu, J.; and Song, L. 2018. Adversarial Attack on Graph Structured Data. In *International Conference on Machine Learning*, 1123–1132.

Decelle, A.; Krzakala, F.; Moore, C.; and Zdeborová, L. 2011. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E* 84(6): 066106.

Diakonikolas, I.; Kamath, G.; Kane, D.; Li, J.; Moitra, A.; and Stewart, A. 2019. Robust Estimators in High-Dimensions Without the Computational Intractability. *SIAM Journal on Computing* 48(2): 742–864.

Feige, U.; and Ofek, E. 2005. Spectral techniques applied to sparse random graphs. *Random Structures & Algorithms* 27(2): 251–275.

Fout, A.; Byrd, J.; Shariat, B.; and Ben-Hur, A. 2017. Protein interface prediction using graph convolutional networks. In *NIPS 2017*, 6530–6539.

Friedman, J. 2004. A proof of Alon's second eigenvalue conjecture and related problems. *Mem. Amer. Math. Soc.* (910).

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. In *ICLR 2015*.

Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. *NIPS 2017* 1024–1034.

Hammond, D. K.; Vandergheynst, P.; and Gribonval, R. 2011. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis* 30(2): 129–150.

Henaff, M.; Bruna, J.; and LeCun, Y. 2015. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163* .

Holland, P. W.; Laskey, K. B.; and Leinhardt, S. 1983. Stochastic blockmodels: First steps. *Social Networks* 5(2): 109–137.

Jacobsen, J.-H.; Smeulders, A.; and Oyallon, E. 2018. i-RevNet: Deep Invertible Networks. In *ICLR 2018*.

Joseph, A.; Yu, B.; et al. 2016. Impact of regularization on spectral clustering. *The Annals of Statistics* 44(4): 1765–1791.

Kaufmann, E.; Bonald, T.; and Lelarge, M. 2016. A spectral algorithm with additive clustering for the recovery of overlapping communities in networks. In *ALT 2016*, 355–370. Springer.

Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; and Riley, P. 2016. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design* 30(8): 595–608.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR 2017*.

Le, C. M.; Levina, E.; and Vershynin, R. 2015. Sparse random graphs: regularization and concentration of the laplacian. *arXiv preprint arXiv:1502.03049* .

Lee, J. B.; Rossi, R. A.; Kong, X.; Kim, S.; Koh, E.; and Rao, A. 2018. Higher-order graph convolutional networks. *arXiv preprint arXiv:1809.07697* .

Lee, J. R.; Gharan, S. O.; and Trevisan, L. 2014. Multiway spectral partitioning and higher-order cheeger inequalities. *Journal of the ACM* 61(6): 37.

Li, Q.; Wu, X.-M.; Liu, H.; Zhang, X.; and Guan, Z. 2019. Label efficient semi-supervised learning via graph filtering. In *CVPR 2019*, 9582–9591.

Li, Z.; Liu, J.; Chen, S.; and Tang, X. 2007. Noise robust spectral clustering. In *ICCV 2007*, 1–8.

Liao, R.; Zhao, Z.; Urtasun, R.; and Zemel, R. S. 2019. LanczosNet: Multi-Scale Deep Graph Convolutional Networks. In *ICLR 2019*.

Lu, Q.; and Getoor, L. 2003. Link-based classification. In *ICML 2003*, 496–503.

Maehara, T. 2019. Revisiting Graph Neural Networks: All We Have is Low-Pass Filters. *arXiv preprint arXiv:1905.09550* .

Marcheggiani, D.; and Titov, I. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *EMNLP 2017*, 1506–1515.

Massoulié, L. 2014. Community detection thresholds and the weak Ramanujan property. In *STOC 2014*, 694–703. ACM.

Mossel, E.; Neeman, J.; and Sly, A. 2012. Stochastic block models and reconstruction. *arXiv preprint arXiv:1202.1499* .

Mossel, E.; Neeman, J.; and Sly, A. 2013. A Proof Of The Block Model Threshold Conjecture. *arXiv preprint arXiv:1311.4115* .

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *KDD 2014*, 701–710. ACM.

Rahimi, A.; Cohn, T.; and Baldwin, T. 2018. Semi-supervised User Geolocation via Graph Convolutional Networks. In *ACL 2018*, volume 1, 2009–2019.

Sun, L.; Wang, J.; Yu, P. S.; and Li, B. 2018. Adversarial Attack and Defense on Graph Data: A Survey. *arXiv preprint arXiv:1812.10528* .

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the Inception Architecture for Computer Vision. In *CVPR 2016*, 2818–2826.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. In *ICLR 2014*.

Weston, J.; Ratle, F.; Mobahi, H.; and Collobert, R. 2012. Deep Learning via Semi-Supervised Embedding. *Neural Networks: Tricks of the Trade (2nd ed.)* 639–655.

Wu, F.; Souza, A. H.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. Q. 2019. Simplifying Graph Convolutional Networks. In *ICML 2019*, 6861–6871.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2019. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596* .

Yang, Z.; Cohen, W. W.; and Salakhutdinov, R. 2016. Revisiting semi-supervised learning with graph embeddings. In *ICML 2016*, 40–48.

Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W. L.; and Leskovec, J. 2018. Graph convolutional neural networks for web-scale recommender systems. In *KDD 2018*, 974–983. ACM.

Zhou, J.; Cui, G.; Zhang, Z.; Yang, C.; Liu, Z.; and Sun, M. 2018. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434* .

Zhu, D.; Zhang, Z.; Cui, P.; and Zhu, W. 2019. Robust Graph Convolutional Networks Against Adversarial Attacks. In *KDD 2019*, 1399–1407.

Zhu, X.; Ghahramani, Z.; and Lafferty, J. D. 2003. Semi-supervised learning using Gaussian fields and harmonic functions. In *ICML 2003*, 912–919.

Zügner, D.; and Günnemann, S. 2019. Adversarial Attacks on Graph Neural Networks via Meta Learning. In *ICLR 2019*.