Interpretable Stability Bounds for Spectral Graph Filters

Henry Kenlay [1] **Dorina Thanou** [2] **Xiaowen Dong** [1]

Abstract

Graph-structured data arise in a variety of real-world context ranging from sensor and transportation to biological and social networks. As a ubiquitous tool to process graph-structured data, spectral graph filters have been used to solve common tasks such as denoising and anomaly detection, as well as design deep learning architectures such as graph neural networks. Despite being an important tool, there is a lack of theoretical understanding of the stability properties of spectral graph filters, which are important for designing robust machine learning models. In this paper, we study filter stability and provide a novel and interpretable upper bound on the change of filter output, where the bound is expressed in terms of the endpoint degrees of the deleted and newly added edges, as well as the spatial proximity of those edges. This upper bound allows us to reason, in terms of structural properties of the graph, when a spectral graph filter will be stable. We further perform extensive experiments to verify intuition that can be gained from the bound.

1. Introduction

A graph is a general-purpose data structure that uses edges to model pairwise interactions between entities, which are modelled as the nodes of the graph. Many types of data in the real-world reside on graph domains, such as those collected in sensor, biological, and social networks. This has sparked a major interest in recent years in developing machine learning models for graph-structured data (Chami et al., 2020), leading to the fast-growing fields of graph signal processing (Shuman et al., 2013; Ortega et al., 2018) and geometric deep learning (Bronstein et al., 2017).

Spectral graph filters, generalisation of classical filters to the graph domain via spectral graph theory (Chung, 1997), are a ubiquitous tool designed to process graph-structured data. In addition to various signal processing tasks (Shuman et al., 2013; Ortega et al., 2018), graph filters are becoming an important tool for machine learning tasks defined on graphs (Dong et al., 2020). For example, they have been used to define convolution on graphs and design graph neural networks, which lead to state-of-the-art performance in both node and graph classification (Bruna et al., 2014; Defferrard et al., 2016; Kipf & Welling, 2017; Levie et al., 2019; Wu et al., 2019; Rossi et al., 2020; Balcilar et al., 2020).

Despite the surge of research proposing new graph-based machine learning models, significantly less attention has been paid to the understanding of theoretical properties, such as stability, of existing models, in particular graph filters. Informally, a filter is considered to be stable against a perturbation if, after being applied to a signal, it does not lead to large changes in the filter output. In the context of graph-structured data, stability can be defined with respect to perturbation to the signal or the underlying topology. We focus on the latter in this work as graph filters are typically through a function of the graph topology.

Stability is important mainly for two reasons. First, real-world graph-structured data often come with perturbations, either due to measurement error or inaccuracy in the graph construction. Second, when these data are used in machine learning tasks, stability of the graph filters is important to designing learning algorithms that are robust to small perturbations. As a practical example, graph filters are often used to extract spatio-temporal predictive features from fMRI signals realised as signals on a structural brain network. The underlying structural brain network is typically an approximation of the true brain connectivity and therefore the topology will inherently be noisy. Nevertheless, we would desire the predictions, and thus the filtering process, to be robust to the noise inherent in this data.

There has only been a handful of papers considering the stability of spectral graph filters. Among the existing works (Levie et al., 2019; Kenlay et al., 2020b; Gama et al., 2020) which address this, most provide an upper bound on the change of the filter output. These upper bounds are in terms of the magnitude of perturbation and lack interpretation in terms of how the structure of the graph has changed, for example the degree of the nodes. This limitation hampers

[1]University of Oxford [2]École Polytechnique Fédérale de Lausanne. Correspondence to: Henry Kenlay <kenlay@robots.ox.ac.uk>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

the design of strategies that could defend efficiently against potential adversaries. A notable exception in the literature is the recent work of Kenlay et al. (2020a); however, this work only considers degree preserving edge rewiring which is a stringent assumption that does not cover many perturbations observed in practical scenarios.

In this work, we provide a novel upper bound for the output change of spectral graph filters under topological perturbation, i.e., edge deletions and additions, of an unweighted and undirected graph. Unlike previous works, our bound is interpretable in the sense that it is expressed in terms of the structural properties of the graph and the perturbation. The bound helps us understand sufficient conditions under which a spectral graph filter will be stable. Specifically, we show that, when edges are deleted and added to a graph to obtain the perturbed graph, the filter will be stable if 1) the endpoints of these edges have sufficiently high degree, and 2) the perturbation is not concentrated spatially around any one node. We further verify the intuition behind our theoretical results using synthetic experiments.

Our study has two main contributions. First, to the best of our knowledge, our theoretical analysis is one of the first that provides sufficient conditions for a graph filter to be robust to the perturbation, where the conditions are in terms of the structural properties rather than the magnitude of change. Second, unlike previous theoretical studies, we perform extensive experiments to validate the intuition gained from the bound. In particular, we examine how the filter output changes for a range of perturbation strategies including random strategies, an adversarial attack strategy and a robust strategy which is derived from insight that the bound provides. Furthermore, we experiment with a range of random graph models as well as real-world data sets, and examine how different properties of these graphs, for example the degree distribution, have an effect on the filter stability. Overall, we believe this study fills an important gap in our understanding of spectral graph filters, and future work based on these ideas can have broad implications for understanding and designing robust graph-based machine learning models that utilise graph filters, most notably a wide range of designs of graph neural networks (Balcilar et al., 2020).

2. Related work

2.1. Stability of graph filters

Stability of graph filters has been mainly studied by characterising the magnitude of perturbation caused by changes to a graph shift operator (GSO) under the operator norm. One such effort is the work of Levie et al. (2019), where filters are shown to be stable in the Cayley smoothness space, with the output change being linearly bounded. The main limitations of this result is that the constant which depends on the filter is not easily interpretable and the bound is only valid for sufficiently small perturbation. In a similar vein, Kenlay et al. (2020b) proves that polynomial graph filters are linearly bounded under changes to the shifted normalised Laplacian matrix by applying Taylor's theorem for matrix functions (Deadman & Relton, 2016). We build upon this work by giving a tighter bound for a larger class of filters, and providing theoretical basis for how the magnitude of change in the Laplacian matrix relates to change in the structural properties of the graph such as the degree of the nodes and the distribution of the perturbed edges.

Studying perturbation with respect to operator norm does not provide invariance to relabelling of nodes. The authors in Gama et al. (2020) address this issue by proposing the Relative Perturbation Modulo Permutation, that considers all permutations of the perturbed GSO. This measure is at least as hard to compute as the graph isomorphism test for which no polynomial time algorithm is known (Babai, 2016). We believe that incorporating permutations may be beneficial in certain cases but not others. For example, the node labelling in polygon meshes is arbitrary and so the distance should be invariant to the labelling. On the other hand, node labelling in social networks corresponds to a user ID and therefore should remain fixed when measuring the change in the graph.

The work most related to ours is by Kenlay et al. (2020a) as it provides structurally interpretable bound. The authors show that under degree preserving edge rewiring the change of spectral graph filters applied to the augmented adjacency matrix depends on the locality of edge rewiring and the degree of the endpoints of the edges being rewired. However, they only consider a specific type of perturbation, which greatly simplifies but at the same time limits the analysis. We consider more general perturbations in this work and derive a similar result as a special case.

2.2. Adversarial attack and defence

Adversarial attacks are an optimisation based data-driven approach to finding perturbations for which graph-based models are not robust. In particular, it has been shown that the output of graph neural networks can change drastically even under small changes to the input generated from adversarial examples (Zügner et al., 2018; Sun et al., 2018). As our bound necessarily covers worst-case scenarios, adversarial attacks provide insight into the tightness of our bound.

Complimentary to this line of work, adversarial defence is concerned with the design of models robust to adversarial attacks. In the context of adversarial defence, our work is tangentially related to certified robustness, whose goal is to theoretically guarantee that some nodes will re-

main correctly classified, e.g., in a semi-supervised node classification task, under small but arbitrary perturbation (Bojchevski & Günnemann, 2019). We are instead interested in certifying what kind of perturbation will lead to small changes in the output of a fixed graph filter.

3. Preliminaries and problem formulation

We define a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a set of n vertices and $\mathcal{E}$ a set of edges. We write $u \sim v$ if node u is connected to v and $u \not\sim v$ otherwise. By fixing a labelling on the nodes we can encode $\mathcal{G}$ into a binary adjacency matrix $\mathbf{A} \in \{0,1\}^{n\times n}$. The degree d_u of a node u indicates the number of nodes connected to u and we define the degree matrix as $\mathbf{D} = diag(d_1, \dots, d_n)$. A node u is said to be isolated if it has degree zero. The normalised Laplacian matrix is defined as $\mathbf{L} = \mathbf{I}_n - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ where $\mathbf{I}_n$ is the identity matrix of dimension n and conventionally the entry $\mathbf{D}_{uu}^{-1/2}$ is set to zero if the node u is isolated. The entries of $\mathbf{L}$ can be explicitly written as

$$\mathbf{L}_{uv} = \begin{cases} 1 & \text{if } u = v \\ \frac{-1}{\sqrt{d_u d_v}} & \text{if } u \sim v \text{ and } u \neq v \\ 0 & \text{otherwise} \end{cases}. \tag{1}$$

The normalised Laplacian matrix is an example of a GSO, a generalisation of the shift operator from classical signal processing which can be used as a building block to construct a graph signal processing framework (Ortega et al., 2018). The matrix $\mathbf{L}$ is real and symmetric, and thus has an eigendecomposition $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ where $\mathbf{\Lambda} = \text{diag}(\lambda_1 \dots \lambda_n)$ are the eigenvalues such that $0 = \lambda_1 \leq \dots \leq \lambda_n \leq 2$ and $\mathbf{U}$ is the matrix where the columns are the corresponding unit norm eigenvectors.

We can define a graph signal $x : \mathcal{V} \to \mathbb{R}$ as an assignment of each node to a scalar value; this can be compactly represented by a vector $\mathbf{x}$ such that $\mathbf{x}_i = x(i)$. The graph Fourier transform can be defined as $\hat{\mathbf{x}} = \mathbf{U}^T\mathbf{x}$ and the inverse graph Fourier transform is then given by $\mathbf{x} = \mathbf{U}\hat{\mathbf{x}}$. With a notion of frequency, filtering signals on graphs amounts to amplifying and attenuating the frequency components in the graph Fourier domain, i.e., $y = \mathbf{U}\,\text{diag}(g(\lambda_1), \dots, g(\lambda_n))\mathbf{U}^T x = \mathbf{U}g(\Lambda)\mathbf{U}^T\mathbf{x} = g(\mathbf{L})\mathbf{x}$, where $g(\cdot)$ is a function over the range of eigenvalues that corresponds to the characteristics of the filter. We will abuse notation by evaluating $g : \mathbb{R} \to \mathbb{R}$ on the domain $\mathbb{R}^{n\times n}$ using this definition of matrix-valued functions[1].

We are primarily concerned with the stability of spectral graph filters where the filter parameters are fixed. This scenario is relevant to hand-tuned filters or during inference of a pre-trained model. An adversarial attack in this setting is known as an evasion attack. Specifically, we consider edge deletions and additions to the graph to give a perturbed graph $\mathcal{G}_p$ and use $\mathbf{L}_p$ to denote the normalised Laplacian of $\mathcal{G}_p$. We consider the magnitude of the error matrix, i.e., $\|\mathbf{E}\|_2 = \|\mathbf{L}_p - \mathbf{L}\|_2$, which we call the error norm where $\|\cdot\|_2$ is the operator norm when applied to matrices and the ℓ_2-norm when applied to vectors. We will also consider the matrix one norm $\|\mathbf{E}\|_1 = \max_i \sum_j |\mathbf{E}_{ij}|$ and the matrix infinity norm $\|\mathbf{E}\|_\infty = \max_j \sum_i |\mathbf{E}_{ij}|$. If we denote $\mathbf{E}_u$ as the row corresponding to node u of $\mathbf{E}$ we can write the matrix one norm as $\|\mathbf{E}\|_1 = \max_u \|\mathbf{E}_u\|_1$ where $\|\cdot\|_1$ is the Manhattan or ℓ_1-norm when applied to vectors.

The goal of this study is two-fold: 1) understand how the relative output of a filter changes when we perturb the topology of the underlying graph; 2) what is the impact of the structural properties of the perturbation on filter stability. In particular, the structural properties we consider are the degree of the nodes before and after perturbation and how much the perturbation is concentrated around each node. We address the first goal in Section 4 and the second in Section 5. We experimentally validate the insights gained by our bound in Section 6.

4. Linearly stable filters

Our notion of stability is based on relative output distance defined as

$$\frac{\|g(\mathbf{\Delta})\mathbf{x} - g(\mathbf{\Delta}_p)\mathbf{x}\|_2}{\|\mathbf{x}\|_2}, \tag{2}$$

where g is a spectral graph filter, $\mathbf{x}$ is an input graph signal and $\mathbf{\Delta}$ is the GSO of the graph $\mathcal{G}$ (similarly $\mathbf{\Delta}_p$ is the GSO of $\mathcal{G}_p$). If we assume $\mathbf{x}$ has unit norm then the above is equivalent to absolute output distance. We can bound this quantity by what we call the filter distance which measures the largest possible relative output change of the filter over non-zero signals:

$$\begin{aligned} \frac{\|g(\mathbf{\Delta})\mathbf{x} - g(\mathbf{\Delta}_p)\mathbf{x}\|_2}{\|\mathbf{x}\|_2} &\leq \max_{\mathbf{x}\neq 0} \frac{\|g(\mathbf{\Delta})\mathbf{x} - g(\mathbf{\Delta}_p)\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \\ &\overset{\text{def}}{=} \|g(\mathbf{\Delta}) - g(\mathbf{\Delta}_p)\|_2. \end{aligned} \tag{3}$$

In Kenlay et al. (2020b), the authors bound the filter distance of a graph filter g where g is a polynomial. The bound is given by some constant times the error norm $\|\mathbf{E}\|_2$, where the constant depends on the filter. When a filter satisfies this property we say it is linearly stable which we define as follows.

Definition 1. *A spectral graph filter $g : \mathbb{R} \to \mathbb{R}$ is said to be linearly stable with respect to a type of graph shift operators, if for any graph shift operators $\mathbf{\Delta}$ and $\mathbf{\Delta}_p$ of this type, we have that*

$$\|g(\mathbf{\Delta}) - g(\mathbf{\Delta}_p)\|_2 \leq C\|\mathbf{E}\|_2 \tag{4}$$

[1] This is one of a few equivalent ways used to define matrix-valued functions (Higham, 2008)

for some positive constant $C \in \mathbb{R}$. The positive constant C is referred to as the stability constant.

Two types of filters of particular interest are the polynomial filters, i.e., $g(\lambda) = \sum_{k=0}^{K} \theta_k \lambda^k$, where $\{\theta_k\}_{k=0}^{K}$ are the polynomial coefficients, and the low-pass filters, i.e., $g(\lambda) = (1+\alpha\lambda)^{-1}$, where $\alpha > 0$ is some constant. Polynomial filters are used in a variety of graph-based machine learning. We list some of them in Table 1. It was recently proved that polynomial filters are linearly stable with respect to the shifted normalised Laplacian matrix $\mathbf{L} - \mathbf{I}_n$ (Kenlay et al., 2020b). A simpler proof with a tighter bound (smaller stability constant) was given to show linear stability with respect to the augmented adjacency matrix $\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$ where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I_n}$ and $\tilde{\mathbf{D}} = \mathbf{D} + \mathbf{I_n}$ (Kenlay et al., 2020a). In addition, the following more general result holds.

Proposition 1. *Polynomial filters $g(\lambda) = \sum_{k=0}^{K} \theta_k \lambda^k$ are linearly stable with respect to any GSO where the spectrum lies in $[-1, 1]$. The stability constant is given by $C = \sum_{k=1}^{K} k|\theta_k|$.*

Proof: See Supplementary Material. Another important class of filters are low-pass filters, which are linearly stable with respect to the normalised Laplacian matrix.

Proposition 2. *The low-pass filter $g(\lambda) = (1+\alpha\lambda)^{-1}$ is linearly stable with respect to the normalised Laplacian matrix. The constant is given by $C = \alpha$.*

Proof: See Supplementary Material. A thorough characterisation of linearly stable filters is beyond the scope of this work. Instead, this section serves to motivate why we are interested in analysing the magnitude of $\|\mathbf{E}\|_2$: some common types of filters are stable to small perturbation when the perturbation is measured by the error norm $\|\mathbf{E}\|_2$. Although this is an intuitive choice, it is not immediately clear how $\|\mathbf{E}\|_2$ is related to the characteristics of the structural properties of the perturbation. This motivates us to provide an upper bound on $\|\mathbf{E}\|_2$ in Section 5 in terms of interpretable characteristics in the structural domain.

5. Interpretable bound on filter output change

In this section we bound $\|\mathbf{E}\|_2$ by interpretable properties relating to the structural change. Given a perturbation and a node u we denote $\mathcal{A}_u$, $\mathcal{D}_u$, and $\mathcal{R}_u$ as the set of adjacent nodes for newly added edges, deleted edges, and remaining edges around u, respectively. We denote $\Delta_u^+ = |\mathcal{A}_u|$ and $\Delta_u^- = |\mathcal{D}_u| < d_u$ the number of edges added and deleted around u, respectively, and $\Delta_u = \Delta_u^+ - \Delta_u^-$ as the change of degree. We denote $d'_u = d_u + \Delta_u$ as the degree of node u in $\mathcal{G}_p$. We define $\alpha_u = \max_{v \in \mathcal{N}_u \cup \{u\}} |\Delta_v|/d_v$, where $\mathcal{N}_u$ is the 1-hop neighbourhood of node u, as the maximum relative change in degree among a node u and its neighbours. In addition, we define $\delta_u = \min_{v \in \mathcal{N}_u} d_v$ as the smallest degree of the nodes neighbouring node u, and δ'_u as the same quantity in the perturbed graph. We assume that both the graph $\mathcal{G}$ and the perturbed graph $\mathcal{G}_p$ do not contain isolated nodes.

Our approach to upper bounding $\|\mathbf{E}\|_2$ relies on the inequality $\|\mathbf{E}\|_2^2 \leq \|\mathbf{E}\|_1 \|\mathbf{E}\|_\infty$ (Higham, 2002, Section 6.3). As $\mathbf{E}$ is Hermitian, $\|\mathbf{E}\|_1 = \|\mathbf{E}\|_\infty$ thus simplifying this inequality to become $\|\mathbf{E}\|_2 \leq \|\mathbf{E}\|_1$. There may exist strategies which give tighter bounds, but the benefit of this approach is that $\|\mathbf{E}\|_1$ leads to an interpretation in the structural domain. Thus, we are making use of the following inequality

$$\|\mathbf{E}\|_2 \leq \|\mathbf{E}\|_1 = \max_{u \in \mathcal{V}} \|\mathbf{E}_u\|_1. \tag{5}$$

By considering how the entries of $\mathbf{L}$ in Eq. (1) change, we have the following closed-form expression for $\|\mathbf{E}_u\|_1$:

$$\begin{aligned}\|\mathbf{E}_u\|_1 = &\sum_{v \in \mathcal{D}_u} \frac{1}{\sqrt{d_u d_v}} + \sum_{v \in \mathcal{A}_u} \frac{1}{\sqrt{d'_u d'_v}} \\ &+ \sum_{v \in \mathcal{R}_u} \left|\frac{1}{\sqrt{d_u d_v}} - \frac{1}{\sqrt{d'_u d'_v}}\right|. \end{aligned} \tag{6}$$

The results of this section bound the three terms in this expression, leading to an overall bound to $\|\mathbf{E}_u\|_1$ hence to $\|\mathbf{E}\|_1$ and $\|\mathbf{E}\|_2$. We proceed by bounding each of the terms in Eq. (6).

5.1. Bounding the error norm

Recall that δ_u is the smallest degree in the neighbourhood of a node u, allowing us to bound the first term in Eq. (6) by replacing d_v with $\delta_u \leq d_v$ in the denominator to give:

$$\sum_{v \in \mathcal{D}_u} \frac{1}{\sqrt{d_u d_v}} \leq \sum_{v \in \mathcal{D}_u} \frac{1}{\sqrt{d_u \delta_u}} = \frac{\Delta_u^-}{\sqrt{d_u \delta_u}}. \tag{7}$$

Similarly, we can bound the second term in Eq. (6) as:

$$\sum_{v \in \mathcal{A}_u} \frac{1}{\sqrt{d'_u d'_v}} \leq \sum_{v \in \mathcal{A}_u} \frac{1}{\sqrt{d'_u \delta_u}} = \frac{\Delta_u^+}{\sqrt{d'_u \delta'_u}}. \tag{8}$$

To bound the third term in Eq. (6), we first introduce the following lemma.

Lemma 1. *Let $\alpha_u \in [0, 1)$. Then the following holds:*

$$\begin{aligned}\sum_{v \in \mathcal{R}_u} \left|\frac{1}{\sqrt{d_u d_v}} - \frac{1}{\sqrt{d'_u d'_v}}\right| &\leq \sum_{v \in \mathcal{R}_u} \left(\frac{\alpha_u}{1-\alpha_u}\right) \frac{1}{\sqrt{d_u d_v}} \\ &\leq \left(\frac{\alpha_u}{1-\alpha_u}\right) \frac{d_u - \Delta_u^-}{\sqrt{d_u \delta_u}}. \end{aligned} \tag{9}$$

Proof: See Supplementary Material. The assumption on α_u can be interpreted as follows. If α_u = 0 then the degree of u and that of all nodes in the neighbourhood of u

Table 1. Examples of linearly stable graph filters used for machine learning.

Filter	Functional form	GSO	Stability constant C	Use
Polynomial filter	$\sum_{k=0}^{K}\theta_k\lambda^k$	$\frac{2\mathbf{L}}{\lambda_{\max}}-\mathbf{I}_n$	$\sum_{k=1}^{K}k\lvert\theta_k\rvert$	Chebynet (Defferrard et al., 2016)
Low-pass filter	$(1+\alpha\lambda)^{-1}$	$\mathbf{L}$	α	Low-pass filtering (Ramakrishna et al., 2020)
Monomial	λ^K	$\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$	K	Simple GCN (Wu et al., 2019)
Identity	λ	$\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$	1	GCN (Kipf & Welling, 2017)

are unchanged, so the third term in Eq. (6) becomes zero. If $\alpha_u \geq 1$, then for some node v in $\mathcal{N}_u \cup \{u\}$ we have $|\Delta_v|/d_v \geq 1$. Notice that for all nodes we have $\Delta_v > -d_v$, since the degree after perturbation δ'_v is strictly positive (recall that we do not allow isolated nodes). Therefore, we must instead have $\Delta_v \geq d_v$ which implies $d'_v \geq 2d_v$. In other words, the assumption $\alpha_u < 1$ means that for all nodes v in $\mathcal{N}_u \cup \{u\}$ we have $d'_v < 2d_v$. This limits large amount of change around low degree nodes. It can be noted that if a perturbation does not alter the degree distribution then $\alpha_u = 0$ for all nodes and the third term in Eq. (6) vanishes. We will consider a particular case of degree preserving perturbation in Section 5.2.

By combining the bounds in Eq. (7) and Eq. (8) with Lemma 1, we can further bound Eq. (6):

$$\|\mathbf{E}_u\|_1 \leq \frac{\Delta_u^-}{\sqrt{d_u\delta_u}} + \frac{\Delta_u^+}{\sqrt{d'_u\delta'_u}} + \left(\frac{\alpha_u}{1-\alpha_u}\right)\frac{d_u-\Delta_u^-}{\sqrt{d_u\delta_u}}. \tag{10}$$

By further combining this bound with Eq. (5), we arrive at our main result.

Theorem 1. *Assume that $\alpha_u \in [0,1)$ holds for all nodes $u \in \mathcal{V}$. Then the following holds:*

$$\|\mathbf{E}\|_2 \leq \max_{u\in\mathcal{V}}\left\{\frac{\Delta_u^-}{\sqrt{d_u\delta_u}}+\frac{\Delta_u^+}{\sqrt{d'_u\delta'_u}}+\left(\frac{\alpha_u}{1-\alpha_u}\right)\frac{d_u-\Delta_u^-}{\sqrt{d_u\delta_u}}\right\} \tag{11}$$

Proof: See Supplementary Material. We will explore the looseness of the bound given in Eq. (10) and Eq. (11) in Section 6. In practice, the bound might be loose but it provides insight into when we expect filters to be stable. We will discuss this insight in Section 5.4. In the following subsection, we will consider a special case where we can produce a bound that is tighter in practice.

5.2. Bounding the error norm under edge rewiring

Degree preserving rewiring is a type of edge rewiring such that the perturbation does not change the original degree distribution (Kenlay et al., 2020a). Given two edges $u \sim v$ and $u' \sim v'$ such that $u \not\sim v', u \not\sim u', v \not\sim v'$ and $v \not\sim u'$, the double edge rewiring operation deletes the two edges and introduces the edges $u \sim u'$ and $v \sim v'$ (see Fig. S2 for an illustration). The perturbation consists of two edge deletions and two edge additions and does not change the degree of any nodes involved. This model of perturbation approximately arises in practical applications, where the capacity of a node is fixed and remains at full load such as in communication networks (Bienstock & Günlük, 1994). In this specific scenario, $\alpha_u = 0, \delta_u = \delta'_u$ and $d_u = d'_u$. Furthermore, we know that $\Delta_u^- = \Delta_u^+ = r_u$ where we define r_u as the number of rewiring operations involved around a node u. Using Theorem 1 we get the following corollary.

Corollary 1. *If the perturbation consists of only double edge rewiring operations then:*

$$\|\mathbf{E}\|_2 \leq \max_{u\in\mathcal{V}}\frac{2r_u}{\sqrt{d_u\delta_u}}. \tag{12}$$

A similar bound has been recently derived to bound the change in feature representations of certain graph neural network architectures (Kenlay et al., 2020a).

5.3. Bounding filter output change

To obtain a full bound on filter output change, we combine together bounds that are developed in previous sections. Consider a spectral graph filter g which is linearly stable with respect to the normalised Laplacian matrix. We then have the following bound for the filter output change:

$$\frac{\|g_\theta(\mathbf{L})\mathbf{x}-g_\theta(\mathbf{L}_p)\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \leq \|g_\theta(\mathbf{L})-g_\theta(\mathbf{L}_p)\|_2 \leq C\|\mathbf{E}\|_2$$
$$\leq C\max_{u\in\mathcal{V}}\left\{\frac{\Delta_u^-}{\sqrt{d_u\delta_u}}+\frac{\Delta_u^+}{\sqrt{d'_u\delta'_u}}+\left(\frac{\alpha_u}{1-\alpha_u}\right)\frac{d_u-\Delta_u^-}{\sqrt{d_u\delta_u}}\right\}. \tag{13}$$

The first inequality is introduced in Eq. (3) which relates relative output distance and filter distance. The second inequality comes from our assumption that the graph filter is linearly stable (Eq. (4)) with a stability constant C. Finally, we can make use of Eq. (11) to establish the third inequality, which provides a structurally interpretable bound on the relative output distance. We discuss interpretations of this result in the following subsection.

5.4. Interpretation of the bound

The bounds given in this section let us reason about sufficient conditions under which a perturbation leads to small change in graph filter output. We can conclude from Eq. (5) that perturbations which cause small changes to $\|\mathbf{E}_u\|_1$ over all nodes u guarantee small change in terms of $\|\mathbf{E}\|_2$. When would $\|\mathbf{E}_u\|_1$ be small for a particular node? If α_u is small ($\alpha_u \approx 0$), then $\alpha_u/(1-\alpha_u) \approx 0$ and $1-\alpha_u/(1-\alpha_u) \approx 1$. Therefore the right hand side of Eq. (10) becomes approximately:

$$\|\mathbf{E}_u\|_1 \approx \frac{\Delta_u^+}{\sqrt{d'_u \delta'_u}} + \frac{\Delta_u^-}{\sqrt{d_u \delta_u}}. \tag{14}$$

This approximation holds, which in turn leads to a small $\|\mathbf{E}_u\|_1$, if we add and delete edges only between nodes with large degrees. The approximation becomes equality in the case when the degree distribution is preserved such as in Section 5.2. When would $\|\mathbf{E}_u\|_1$ be small for all nodes? Intuitively, this requires perturbations to be distributed across the graph, i.e., not concentrated around any one node. Therefore, spectral graph filters are most robust when we a) add or delete edges between high degree nodes, and b) do not perturb too much around any one node. In the next section, we empirically verify the looseness of each of the bounds and the intuition it provides.

6. Experiments

We empirically verify the looseness of the bounds derived in the previous section. We perform an extensive study of the looseness of these bounds by considering a variety of experimental conditions in terms of different graph types (both random graph models and real-world graph data) and perturbation strategies. Clearly, as the overall bound in Eq. (13) is obtained from a chain of inequalities, its looseness is affected by the looseness of each individual bound (Eqs. (3), (4), (5), (10), (11)). For completeness, the looseness of the inequalities relating the relative output distance and the filter distance (Eq. (3)) is illustrated in Fig. S3, and that of the inequality relating the filter distance and the constant times the error norm (Eq. (4)) in Fig. S4.

6.1. Experimental setup

We generate synthetic graphs on 100 nodes, using different random graph models. With the exception of the stochastic block model and the real-world data, we generate features on the nodes of the graph by taking a random convex combination of the first 10 eigenvectors of the normalised graph Laplacian. The latter results in relatively smooth signals on the graph. For the stochastic block model, we generate graphs with three equal-size communities generating each community's features using a Gaussian with means 2, 0, and -2 respectively and unit variance. For the real-world data we select a continuous feature channel and normalise by subtracting the mean and dividing by the standard deviation. Gaussian noise is then added to generate noisy signals at a signal-to-noise ratio of 0 dB (equal levels of signal and noise).

For the sake of simplicity, we focus in this paper on a fixed low-pass filter $g(\lambda) = (1+\lambda)^{-1}$, which has been widely used for signal denoising (Ramakrishna et al., 2020) and semi-supervised learning (Belkin et al., 2004) tasks. This filter has stability constant $C = 1$ and thus satisfies the inequality $\|g(\mathbf{L}) - g(\mathbf{L}_p)\|_2 \leq \|\mathbf{E}\|_2$, due to Proposition 2. We note though that the experiments may be extended to any type of linearly stable graph filter. We compare the filtering outcome before and after perturbation to the graph topology in a signal denoising task. To this end, we are interested in bounding the relative output distance between the denoised signal before and after perturbation, for different random graph models and perturbation strategies. The magnitude of the perturbation is set at $\lfloor 10\% \cdot |\mathcal{E}| \rfloor$ edge edits. Each experiment is repeated 100 times using different random seeds. We consider varying the experimental settings such as the size of the graphs, amount of the perturbation and level of noise in Supplementary Material.

We note that in some experiments the assumption of Lemma 1 is not satisfied, i.e., for some node u it holds that $\alpha_u \geq 1$. We call an experiment valid if the assumption holds for all nodes. The validity of an experiment depends on both the strategy of perturbation and the type of graph used (Table S2). We only report results from valid experiments for plots directly related to the bounds given in Lemma 1 or Theorem 1. We discuss this further in Supplementary Material.

We use a variety of random graph models including the Erdős-Rényi model (ER), Barabási–Albert model (BA), Watts–Strogatz model (WS), K-regular graphs (K-Reg), Stochastic Block Model (SBM), and assortative graphs (Assortative) (Barabási, 2013, Chapter 7). The random graph models include graphs with low variance (WS and K-Reg) and high variance (SBM, ER, Assortative and BA) in degree distribution. We also consider two real-world data sets, namely PROTEINS_full and ENZYMES. To control for graph size we used 100 graphs with n from 40 to 50 for ENZYMES and 50 to 75 for PROTEINS_full. The standard deviation of the node degrees averaged across the 100 graphs were 1.01 for ENZYMES and 1.03 for PROTEINS_full, similar to synthetic graphs with low degree variance. The standard deviation of the degree distribution averaged over each graph type is given in Table S1. Further details of the random graph models are found in Supplementary Material.

The perturbation strategies under consideration, for a fixed budget, are as follows: 1) randomly selecting edges to delete (Delete); 2) randomly selecting edges to add (Add); 3) us-

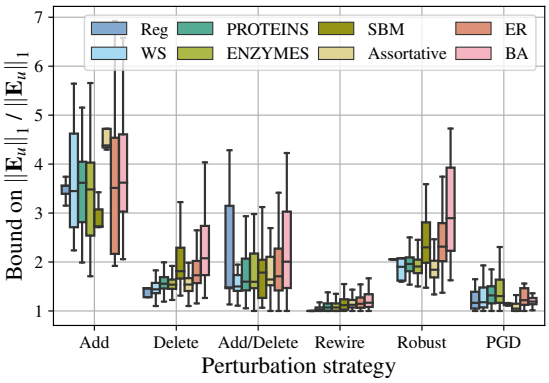

Figure 1. Looseness of the bound given in Eq. (10).

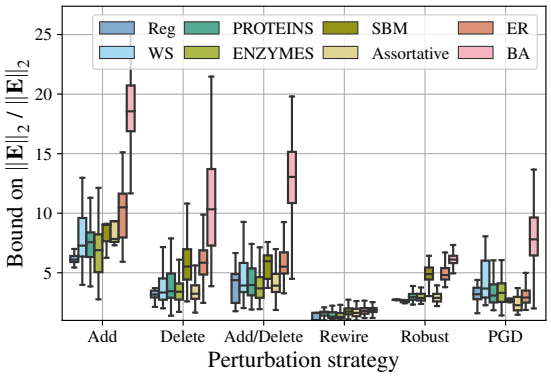

Figure 2. Looseness of the bound given in Eq. (11).

ing half the budget to randomly add and half to randomly delete (Add/Delete); 4) using degree preserving double edge rewiring as described in Section 5.2 (Rewire), which is a special case of Add/Delete (we consider a single double edge rewiring to be four edits, i.e., two deletions and two additions); 5) projected gradient descent (PGD), which is used to find adversarial examples by perturbing the graphs similarly to that described in Xu et al. (2019); and 6) sequentially deleting or adding edges in a greedy manner to minimise $\|\mathbf{E}\|_1$ (Robust). Further details of the perturbation strategies are described in Supplementary Material.

6.2. How tight is the bound $||\mathbf{E}||_2 \leq ||\mathbf{E}||_1$?

We upper bound $\|\mathbf{E}\|_2$ using the inequality given in Eq. (5). In order to quantify the tightness of the bound, we compare in Fig. S5 the values of $\|\mathbf{E}\|_1$ and $\|\mathbf{E}\|_2$ for different perturbation strategies, by illustrating their correlation. We note that Robust leads to the smallest values of $\|\mathbf{E}\|_2$ among all perturbation strategies. This is expected as Eq. (5) tells us that small values of $\|\mathbf{E}\|_1$, which are achieved with the Robust perturbation strategy, guarantee small values of $\|\mathbf{E}\|_2$. As a matter of fact, we observe experimentally that the two norms are correlated ($r = 0.90$), confirming that in practice if we observe $\|\mathbf{E}\|_2$ to be small then $\|\mathbf{E}\|_1$ is likely to be small too.

We show in Fig. S6 the looseness of the bound given in Eq. (5) among the different perturbation strategies and graph models[2]. We see that the bound is tightest for Robust and Rewire. It is interesting to observe that Rewire sometimes gives a tight bound for the graphs with low variance in degree distribution.

6.3. How tight are the bounds on $||\mathbf{E}||_1$ and $||\mathbf{E}||_2$?

We now turn our attention to the bound given in Section 5. As well as calculating the overall value of $||\mathbf{E}_u||_1$ we can also compute the contributions of the three terms in Eq. (6). This allows us to evaluate the looseness of each term as well as the overall looseness of Eq. (10). For each experiment we selected the node u such that $u = \arg\max \|\mathbf{E}_u\|_1$ and calculated the terms and bounds for this node. The results for each term are shown in Fig. S7 and that for the overall looseness in Fig. 1.

As one can see from Fig. S7, the bounds for particular terms are tight in certain scenarios. For example, the inequality in Eq. (7) becomes equality when $d_v = \delta_u$ for all neighbouring nodes v of u, which is the case for 3-Reg graphs. The inequality in Eq. (7) and Eq. (8) is loosest when $\delta_u \ll d_v$ for many neighbouring nodes v. This is likely to occur when the degree distribution has high variance, explaining why the bound for the first and second term are looser for graphs with higher variance in degree distribution. The bound on the third term is the loosest in practice. From Fig. 1, the overall bound is tightest for the Rewire strategy, and this is because both the third term and the bound for it are zero in this case.

Fig. 2 shows the looseness of the bound on $||\mathbf{E}||_2$ in Eq. (11). The overall pattern is similar to that in Fig. 1, where the bound is tight in some rewiring experiments. In general the bound performs poorly on BA graphs, likely due to the skewed degree distribution.

6.4. When are filters robust?

In this section we take a holistic view of the bound in Eq. (13), considering how the relative output distance is affected by the perturbation strategy, graph model, and the statistics that appear throughout our chain of inequalities. We use insight from our bounds to demonstrate scenarios where a filter is robust. Fig. 3 shows how the different graphs and strategies effect each of the quantities that appear in our bounds. In many cases, Robust gives overall the smallest values where PGD gives the largest, which is expected due to the nature of these strategies. It is interesting to note that

[2]We do not display outliers for figures that appear in main text to make the rest of the data easier to visualise.

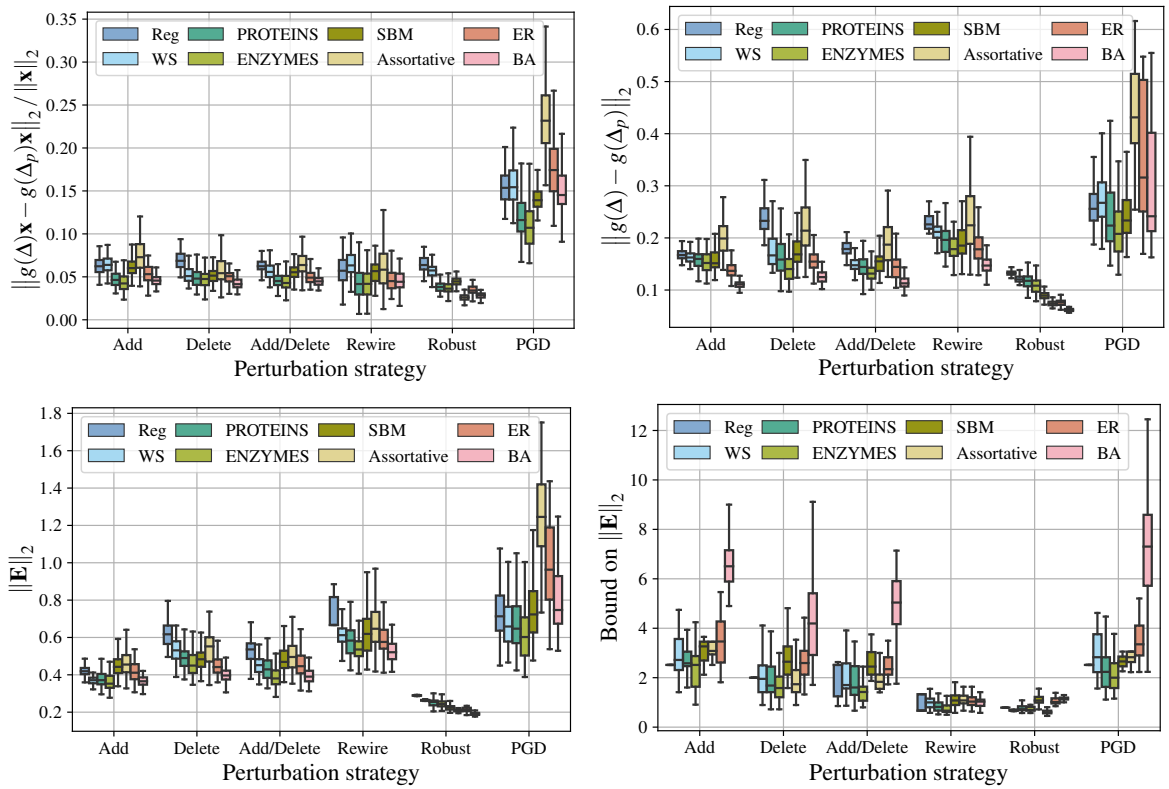

Figure 3. How different statistics vary across experimental setups.

the Robust strategy gives smallest relative output distance for synthetic graphs with high degree variance (ER, Assortative, BA and SBM), as well as the real-world data sets, but not for those with low degree variance (WS, 3-Reg). When the degree distribution is flat the strategies perturb edges with similar endpoint degrees due to the lack of choice. On the other hand, the PGD strategy gives larger changes to graphs with higher variance in degree distribution, suggesting the existence of low degree nodes or non-uniform degree distributions that are more vulnerable to adversarial attacks.

In summary, our bounds suggest that filters are robust when we modify edges where the endpoint degrees are high and that the perturbation is distributed across the graph. BA graphs of n nodes have a small diameter that grows asymptotically $\mathcal{O}(\log n / \log \log n)$ (Bollobás & Riordan, 2004). Consequentially, in our experiments on small BA graphs, most edges that are added and deleted are in close proximity. Furthermore, BA graphs have a power-law degree distributions. This type of graph model allows us to control for the distribution of the perturbation and observe instead how the endpoint degrees change across strategies. One can see how PGD tends to target small-degree nodes whereas Robust targets edges connected to the large-degree hubs (Fig. 4).

We finally control for the degree distribution by considering K-regular graphs. In Fig. 5 we can see that Robust

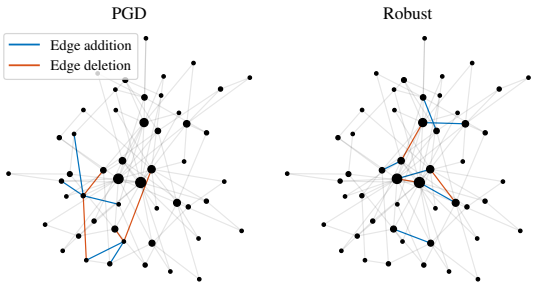

Figure 4. Perturbations of BA graphs ($n = 50$). The original and both perturbed graphs have a diameter of 5. The size of the node is proportional to the node degree.

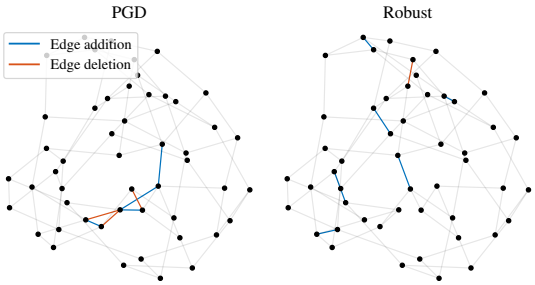

Figure 5. Perturbations of 3-regular graphs ($n = 50$).

deletes and adds edges between nodes in a distributed manner, whereas PGD tends to add edges that are adjacent to each other. This verifies the insight from the bound in terms of robustness with respect to spatial distribution of perturbation.

7. Discussion

In this work, we develop novel interpretable bounds to help elucidate certain types of perturbations against which spectral graph filters are robust. We show that filters are robust when we modify edges where the endpoint degrees are high, and the perturbation is distributed across the graph. Although these bounds are likely to be loose in practice, they provide qualitative insight which we validate through extensive experiments.

We believe that our work can be used in future research to investigate further the stability of graph-based machine learning algorithms. Studying additional perturbation models beyond edge deletion/addition, relaxing the assumption on α_u, allowing nodes to become isolated, and considering perturbation to node features, may all increase the applicability of the framework to practical scenarios. Further statistical investigation into how much the role of degree and edge locality effect stability is an important future direction. Considering weighted graphs is another natural extension of the proposed bounds. Our study is limited to a single fixed filter operating on a particular class of graph shift operator. Extension to a wide range of graph-based machine learning models that might contain multiple spectral graph filters as building blocks is a clear avenue for future research. One such example are graph neural networks, where understanding the stability with respect to perturbations might have positive implications for designing more robust architectures.

References

Albert, R. and Barabási, A.-L. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

Babai, L. Graph isomorphism in quasipolynomial time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pp. 684–697, 2016.

Balcilar, M., Renton, G., Héroux, P., Gauzere, B., Adam, S., and Honeine, P. Bridging the gap between spectral and spatial domains in graph neural networks. *arXiv preprint arXiv:2003.11702*, 2020.

Barabási, A.-L. Network science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1987):20120375, 2013.

Belkin, M., Matveeva, I., and Niyogi, P. Tikhonov regularization and semi-supervised learning on large graphs. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004.

Bienstock, D. and Günlük, O. A degree sequence problem related to network design. *Networks*, 24(4):195–205, 1994.

Bojchevski, A. and Günnemann, S. Certifiable robustness to graph perturbations. In *Advances in Neural Information Processing Systems*, pp. 8319–8330, 2019.

Bollobás, B. and Riordan, O. The diameter of a scale-free random graph. *Combinatorica*, 24(1):5–34, 2004.

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4): 18–42, 2017.

Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. Spectral networks and deep locally connected networks on graphs. In *International Conference on Learning Representations*, 2014.

Chami, I., Abu-El-Haija, S., Perozzi, B., Ré, C., and Murphy, K. Machine learning on graphs: A model and comprehensive taxonomy. *arXiv preprint arXiv:2005.03675*, 2020.

Chung, F. *Spectral graph theory*. American Mathematical Society, 1997.

Deadman, E. and Relton, S. D. Taylor's theorem for matrix functions with applications to condition number estimation. *Linear Algebra and its Applications*, 504:354–371, 2016.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pp. 3844–3852, 2016.

Dong, X., Thanou, D., Toni, L., Bronstein, M., and Frossard, P. Graph signal processing for machine learning: A review and new perspectives. *IEEE Signal Processing Magazine*, 37(6):117–127, 2020.

Gama, F., Bruna, J., and Ribeiro, A. Stability properties of graph neural networks. *IEEE Transactions on Signal Processing*, 68:5680–5695, 2020.

Gilbert, E. N. Random graphs. *The Annals of Mathematical Statistics*, 30(4):1141–1144, 1959.

Higham, N. J. *Accuracy and stability of numerical algorithms*. Society for Industrial and Applied Mathematics, 2002.

Higham, N. J. *Functions of matrices: theory and computation*. SIAM, 2008.

Kenlay, H., Thanou, D., and Dong, X. On the stability of graph convolutional neural networks under edge rewiring. *arXiv preprint arXiv:2010.13747*, 2020a.

Kenlay, H., Thanou, D., and Dong, X. On the stability of polynomial spectral graph filters. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5350–5354, 2020b.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Levie, R., Isufi, E., and Kutyniok, G. On the transferability of spectral graph filters. *arXiv preprint arXiv:1901.10524*, 2019.

Levie, R., Monti, F., Bresson, X., and Bronstein, M. M. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2019.

Ortega, A., Frossard, P., Kovačević, J., Moura, J. M., and Vandergheynst, P. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828, 2018.

Ramakrishna, R., Wai, H. T., and Scaglione, A. A user guide to low-pass graph signal processing and its applications: Tools and applications. *IEEE Signal Processing Magazine*, 37(6):74–85, 2020.

Rossi, E., Frasca, F., Chamberlain, B., Eynard, D., Bronstein, M., and Monti, F. Sign: Scalable inception graph neural networks. *arXiv preprint arXiv:2004.11198*, 2020.

Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., and Vandergheynst, P. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013.

Spivak, M. *Calculus on manifolds: a modern approach to classical theorems of advanced calculus*. CRC press, 2018.

Steger, A. and Wormald, N. C. Generating random regular graphs quickly. *Combinatorics, Probability and Computing*, 8(04):377–396, 1999.

Sun, L., Dou, Y., Yang, C., Wang, J., Yu, P. S., and Li, B. Adversarial attack and defense on graph data: A survey. *arXiv preprint arXiv:1812.10528*, 2018.

Watts, D. J. and Strogatz, S. H. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., and Weinberger, K. Simplifying graph convolutional networks. In *International Conference on Machine Learning*, pp. 6861–6871, 2019.

Xu, K., Chen, H., Liu, S., Chen, P., Weng, T., Hong, M., and Lin, X. Topology attack and defense for graph neural networks: An optimization perspective. 2019.

Xulvi-Brunet, R. and Sokolov, I. M. Reshuffling scale-free networks: From random to assortative. *Physical Review E*, 70(6):066102, 2004.

Zügner, D., Akbarnejad, A., and Günnemann, S. Adversarial attacks on neural networks for graph data. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2847–2856, 2018.