Stability and Generalization of ℓ_p-Regularized Stochastic Learning for GCN

Shiyu Liu[1] , **Linsen Wei**[2] , **Shaogao Lv**[3†] and **Ming Li**[4]
[1]University of Electronic Science and Technology of China, China
[2]School of Astronautics, Northwestern Polytechnical University, China
[3]Department of Statistics and Data Science, Nanjing Audit University, China
[4]Key Laboratory of Intelligent Education Technology and Application of Zhejiang Province, Zhejiang Normal University, China
shyu.liu@foxmail.com, linsenlinwei@gmail.com, lvsg716@nau.edu.cn, mingli@zjnu.edu.cn

Abstract

Graph convolutional networks (GCN) are viewed as one of the most popular representations among the variants of graph neural networks over graph data and have shown powerful performance in empirical experiments. That ℓ_2-based graph smoothing enforces the global smoothness of GCN, while (soft) ℓ_1-based sparse graph learning tends to promote signal sparsity to trade for discontinuity. This paper aims to quantify the trade-off of GCN between smoothness and sparsity, with the help of a general ℓ_p-regularized $(1 < p \leq 2)$ stochastic learning proposed within. While stability-based generalization analyses have been given in prior work for a second derivative objectiveness function, our ℓ_p-regularized learning scheme does not satisfy such a smooth condition. To tackle this issue, we propose a novel SGD proximal algorithm for GCNs with an inexact operator. For a single-layer GCN, we establish an explicit theoretical understanding of GCN with the ℓ_p-regularized stochastic learning by analyzing the stability of our SGD proximal algorithm. We conduct multiple empirical experiments to validate our theoretical findings.

1 Introduction

Graph Neural Networks (GNNs) have emerged as a family of powerful model designs for improving the performance of neural network models on graph-structured data. GNNs have delivered remarkable empirical performance from a diverse set of domains, such as social networks, knowledge graphs, and biological networks [Duvenaud *et al.*, 2015; Battaglia *et al.*, 2016; Defferrard *et al.*, 2016; Jin *et al.*, 2018; Barrett *et al.*, 2018; Yun *et al.*, 2019; Zhang *et al.*, 2020]. In fact, GNNs can be viewed as natural extensions of conventional machine learning for any data where the available structure is given by pairwise relationships.

The architecture designs of various GNNs have been motivated mainly by spectral domain [Defferrard *et al.*, 2016; Kipf and Welling, 2016a] and spatial domain [Hamilton *et al.*, 2017; Gilmer *et al.*, 2017]. Some popular variants of graph neural networks include Graph Convolutional Network (GCN) [Bruna *et al.*, 2013], GraphSAGE [Hamilton *et al.*, 2017], Graph Attention Network [Veličković *et al.*, 2017], Graph Isomorphism Network [Xu *et al.*, 2018], and among others.

Specifically, inherited excellent performances of traditional convolutional neural networks in processing image and time series, a standard GCN [Kipf and Welling, 2016a] also consists of a cascade of layers, but operates directly on a graph and induces embedding vectors of nodes based on properties of their neighborhoods. Formally, GCN is defined as the problem of learning filter parameters in the graph Fourier transform. GCNs have shown superior performances on real datasets from various domains, such as node labeling on social networks [Kipf and Welling, 2016b], link prediction in knowledge graphs [Schlichtkrull *et al.*, 2018], and molecular graph classification in quantum chemistry [Gilmer *et al.*, 2017].

Notably, a recent study [Ma *et al.*, 2021] has proven that GCN, even for general messaging passing models, intrinsically performs the ℓ_2-based graph smoothing signal, which enforces smoothness globally, and the level and smoothness are often shared across the whole graph. As opposed to ℓ_2-based graph smoothing, ℓ_1-based methods tend to penalize large values less and thus preserve discontinuity of non-smooth signal better [Nie *et al.*, 2011; Wang *et al.*, 2015; Liu *et al.*, 2021]. Essentially, ℓ_1-based methods are equivalent to soft-thresholding operations for iterative estimators and guarantee statistical properties (e.g., model selection consistency). Owning to these advantages, trend filtering [Tibshirani, 2014], and graph trend filter [Wang *et al.*, 2015; Verma and Zhang, 2019] indicate that ℓ_1-based graph smoothing can adapt to the inhomogeneous level of smoothness of signals and yield estimator with k-th piecewise polynomial functions, such as piecewise constant, linear and quadratic functions, depending on the order of the graph difference operator.

To enhance the local smoothness adaptivity of GCNs, a family of elastic-type GCNs with a combination of ℓ_2 and ℓ_1-based penalties are proposed by [Liu *et al.*, 2021], which demonstrate that the elastic GCNs obtain better adaptivity on benchmark datasets and are significantly robust to graph adversarial attacks.

Under the regularized learning framework, this paper further studies a class of ℓ_p-regularized learning approaches $(1 < p \leq 2)$, in order to trade-off the local smoothness of

†Corresponding author.

GCNs and sparsity between nodes. To be precise, it has been shown that an extreme case for ℓ_p-regularized learning with $p \to 1$ tends to generate soft-sparsity of solutions [Koltchinskii, 2009]. In analogy to elastic-type GCNs, general ℓ_p-regularization can be interpreted as an interpolation of ℓ_1-regularization and ℓ_2-regularization. However, in contrast with the elastic-type GCNs, ℓ_p-regularized GCNs only involve a regularized parameter but leads to some additional technical difficulties in optimization and theoretical analysis.

The generalization performance of an algorithm has always been a central issue in learning theory, and particularly the generalization guarantees of GNNs has attracted a considerable amount of attention in recent years [Verma and Zhang, 2019; Garg *et al.*, 2020; Liao *et al.*, 2020; Oono and Suzuki, 2020].

It is worth noting that, although ℓ_1-regularization possesses a number of attractive properties such as "automatic" variable selection, the objectiveness with $p = 1$ is not a strictly convex smooth function, which has been proved not to be uniform stability [Xu *et al.*, 2011]. On the other hand, it is known that for $p > 1$, the penalty is a strictly convex function over bounded domains and thus enjoys robustness to some extent. As a bridge of sparse ℓ_1-regularization and dense ℓ_2-regularization, ℓ_p-based learning allows us to explicitly observe a trade-off between sparsity and smoothness of the estimated learners.

Although this idea is natural within the framework of regularized learning, it still faces many technical challenges when applied to GCNs. First, to address the problem of uniform stability of an SGD algorithm that can induce generalization performance [Bousquet and Elisseeff, 2002], existing related analysis for SGD [Verma and Zhang, 2019; Hardt *et al.*, 2016] required that the first derivative of the objectiveness is Lipschitz continuous, which is not applicable to ℓ_p-based GCN. Second, to derive an interpretable generalization bound, it is important to know how such a result depends on the structure of the graph filter and the regularized parameter p, as well as the network size and the sample size.

Overall, our principal contributions can be summarized as follows:

- We introduce ℓ_p-regularized learning approaches for one-layer GCN to provide an explicit theoretical characterization of the trade-off between local smoothness and sparsity of the SGD algorithm for GCN. Crucially, we analyze how this trade-off between the graph structure and the regularization parameter p affects the generalization capacity of our SGD method.
- We propose a novel regularized stochastic algorithm for GCN, i.e., *Inexact Proximal SGD*, by integrating the standard SGD projection and the proximal operator. For our proposed method, we derive interpretable generalization bounds in terms of the graph structure, the regularized parameter p, the network width, and the sample size.
- To our knowledge, we are the first to analyze the generalization performance of ℓ_p-SGD in GNNs, which is quite different from existing stability-based generalization analysis for a second derivative objectiveness [Verma and Zhang, 2019; Hardt *et al.*, 2016]. We also have to overcome additional challenges in understanding the nature of the stochastic gradient for GCN, posed by the message passing nature in GCN. We conduct several numerical experiments to illustrate the superiority of our method to traditional smooth-based GCN, and we also observe some sparse solutions through our experiments as p is sufficiently close to 1.

1.1 Additional Related Work

In this subsection, we briefly review two kinds of related works: Generalization Analysis for GNNs and Regularized Schemes on GNNs.

Generalization Analysis for GNNs. Some previous work has attempted to address the generalization guarantees of GNNs, including [Verma and Zhang, 2019; Garg *et al.*, 2020; Liao *et al.*, 2020; Oono and Suzuki, 2020]. However, most of these works established uniform convergence results using classical Rademacher complexity[Garg *et al.*, 2020; Oono and Suzuki, 2020] and PAC bounds [Liao *et al.*, 2020]. Compared to these abstract capacity notations, the stability concept, used recently for GNN in [Verma and Zhang, 2019], is more intuitive and directly defined over a specific algorithm. Although the stability for generalization of GNNs has been considered recently by [Verma and Zhang, 2019], a major difference from their work is that we focus on the trade-off between soft sparsity and generalization of the ℓ_p-based GCN with varying $p \in (1, 2]$. Moreover, we propose a new SGD algorithm for GCN, under which we provide novel theoretical proof for the stability bound of our SGD.

Regularized Schemes on GNNs. Regularization methods are frequently used in machine learning, especially the prior literature [Wibisono *et al.*, 2009] has studied the influence of ℓ_p-regularized learning on generalization ability. Note that the previous work only considered the impact of general regularization estimates on stability without a specific algorithm. In addition, their research is based on regular data and does not involve any graph structure. As far as we know, this paper is the first time to consider ℓ_p-regularized learning in a graph model.

2 Preliminaries and Methodology

In this section, we first describe basic notation on graph and standard versions of GCN. Then we introduce the structural empirical risk minimization for a single-layer GCN model under i.i.d. sampling process, and thus our ℓ_p-regularized approaches is naturally formulated to estimate the graph filter parameters.

A graph is represented as $G = (V, E)$, where $V = \{\nu_1, \nu_2, ..., \nu_n\}$ is a set of $n = |V|$ nodes and E is a set of $|E|$ edges. The adjacency matrix of the graph is denoted by $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{n\times n}$, whose entries $a_{ij} = 1$ if $(\nu_i, \nu_j) \in E$, and $a_{ij} = 0$ otherwise. Each node's own feature vector is denoted by $\mathbf{x}_i \in \mathbb{R}^d$, $i \in [n]$, where d is the dimension of the node feature. Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ denote the node feature matrix with each row being d features. The 1-hop neighborhood of a node ν_i is defined as the set $\{\nu_j, (\nu_i, \nu_j) \in E\}$, and denote by $N_{(i)}$ the set that includes the node ν_i and all nodes belonging to its 1-hop neighborhood. The main task of graph models is to combine the feature information and the edge information to perform learning tasks.

For an undirected graph, its Laplacian matrix $\mathbf{L} \in \mathbb{R}^{n\times n}$ is defined as $\mathbf{L} := \mathbf{D}-\mathbf{A}$, where $\mathbf{D} \in \mathbb{R}^{n\times n}$ is a degree diagonal matrix whose diagonal entry $d_{ii} = \sum_j a_{ij}$ for $i \in [n]$. The semi-definite matrix $\mathbf{L}$ has an eigen-decomposition written by $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where the columns of $\mathbf{U}$ are the eigenvectors of $\mathbf{L}$ and the diagonal entries of diagonal matrix $\mathbf{\Lambda}$ are the non-negative eigenvalues of $\mathbf{L}$.

For a fixed function g, we define a graph filter $g(\mathbf{L}) \in \mathbb{R}^{n\times n}$ as a function on the graph Laplacian $\mathbf{L}$. Following the eigen-decomposition of $\mathbf{L}$, we get $g(\mathbf{L}) = \mathbf{U}g(\mathbf{\Lambda})\mathbf{U}^T$, where the eigenvalues are given by $\lambda_i^{(g)} = \{g(\lambda_i), 1 \le i \le n\}$. We define $\lambda_G^{max} = \max\{|\lambda_i^{(g)}|\}$ as the largest absolute eigenvalue of the graph filter $g(\mathbf{L})$.

In this paper, we are concerned with node-level semi-supervised learning problems over the graph G. Let $\mathcal{X} \subset \mathbb{R}^d$ be the input space in which the node feature is well defined, and accordingly $\mathcal{Y} \subset \mathbb{R}$ be the output space. In the semi-supervised setting, one assumes that only a portion of the training samples are labeled while amounts of unlabeled data are collected easily. Precisely, we merely collect a training set with labels $D = \{\mathbf{z}_i = (\mathbf{x}_i, y_i)\}_{i=1}^m$ with $m \ll n$. For statistical inference, one often assumes that these pairwise sample are independently drawn from a joint distribution ρ defined on $\mathcal{X} \times \mathcal{Y}$. In such case, our studied model belongs to node-focused tasks on graph, as opposed to graph-focused tasks where the whole graph can be viewed as a single sample.

The most simple graph neural network, known as the Vanilla GCN, was proposed in [Kipf and Welling, 2016a], where each layer of a multilayer network is multiplied by the graph filter before applying a nonlinear activation function. In a matrix form, a conventional multi-layer GCN is represented by a layer-wise propagation rule

$$\mathbf{H}^{(k+1)} = \sigma\big(g(\mathbf{L})\mathbf{H}^{(k)}\mathbf{W}^{(k+1)}\big), \tag{1}$$

where $\mathbf{H}^{(k+1)} \in \mathbb{R}^{n\times m_{k+1}}$ is the node feature representation output by the $(k+1)$-th GCN layer, and specially $\mathbf{H}^{(0)} = \mathbf{X}$ and $m_0 = d$. $\mathbf{W}^{(k+1)} \in \mathbb{R}^{m_k\times m_{k+1}}$ represents the estimated weight matrix of the $(k+1)$-th GCN layer, and σ is a point-wise nonlinear activation function. Under the context of GCN for performing semi-supervised learning, the sampling procedure of nodes from the graph G is conducted by two stages. We assume node data are sampled in an i.i.d. manner by first choosing a sample $\mathbf{x}_i$ or $\mathbf{z}_i$ at node i, and then extracting its neighbors from G to form an ego-graph.

To interpret learning mechanism clearly for GCN, this paper focuses on a node-level task over graph with a single layer GCN model. In such case, putting all graph nodes together, the output function can be written in a matrix form as follows,

$$f(\mathbf{X}, \mathbf{w}) = \sigma\big(g(\mathbf{L})\mathbf{X}\mathbf{w}\big), \tag{2}$$

where $\mathbf{w} \in \mathbb{R}^d$ and $f(\mathbf{X}, \mathbf{w}) \in \mathbb{R}^n$. Some commonly used graph filters include a linear function of $\mathbf{A}$ as $g(\mathbf{A}) = \mathbf{A} + \mathbf{I}$ [Xu *et al.*, 2018] or a Chebyshev polynomial of $\mathbf{L}$ [Defferrard *et al.*, 2016].

Under the context of ego-graph, each node contains the complete information needed for computing the output of a single layer GCN model. Given node with the feature $\mathbf{x}$, let $N_{\mathbf{x}}$ denote a set of the neighbor indexes at most 1-hop distance neighbors, which is completely determined by the graph filter $g(\mathbf{L})$. Thus we can rewrite the predictor (2) for a single node prediction as

$$f(\mathbf{x}, \mathbf{w}) = \sigma\big(\sum\nolimits_{j\in N_{\mathbf{x}}} e_{\cdot j}\mathbf{x}_j^T\mathbf{w}\big), \tag{3}$$

where $e_{\cdot j} = [g(\mathbf{L})]_{\cdot j} \in \mathbb{R}$ is regraded as a weighted edge between node $\mathbf{x}$ and its neighbor $\mathbf{x}_j$, and particularly it still holds $j \in N_{\mathbf{x}}$ if and only if $e_{\cdot j} \neq 0$.

Let $\ell(\cdot,\cdot)$ be a convex loss function, measuring the difference between a predictor and the true label, a variety of supervised learning problems in machine learning can be formulated as a minimization of the expectation risk,

$$\min_{f\in\mathcal{F}} \mathbb{E}\big[\ell\big(Y, f(X)\big)\big], \tag{4}$$

where $\mathcal{F}$ is a hypothesis space under which an optimal learning rule is generated. The standard regression problems correspond to the square loss given by $\ell(u, v) = (u - v)^2$, and the logistic loss is widely used for classification. The optimal decision function denoted by f_0 is any minimizer of (4) when $\mathcal{F}$ is taken to be the space of all measurable functions. However, f_0 can not be computed directly, due to the fact that ρ is often unknown and $\mathcal{F}$ is too complex to compute it possibly. Instead, a frequently used method consists of minimizing a regularized empirical risk over a computationally-feasible space

$$\min_{f\in\mathcal{H}} \frac{1}{m}\sum\nolimits_{i=1}^m \ell\big(y_i, f(\mathbf{x}_i)\big) + \Omega(f), \tag{5}$$

where $\Omega(f)$ is a penalty function that regularizes the complexity of the function f, while $\mathcal{H}$ is the space of all predictors which needs to be parameterized explicitly or implicitly. For instance, the neural network is known as an efficient parameterized approximation to any complex nonlinear function. In the work, all the functions within $\mathcal{H}$ consist of the form in (3).

2.1 Empirical Risk Minimization with ℓ_p-Regularizer

A fundamental class of learning algorithms can be described as the regularized empirical risk minimization problems. This paper considers an ℓ_p-regularized learning approach for training the parameters of GCN:

$$\widehat{\mathbf{w}} \in \arg\min_{\mathbf{w}} \frac{1}{n}\sum\nolimits_{i=1}^n \ell\big(y_i, f(\mathbf{x}_i, \mathbf{w})\big) + \lambda\|\mathbf{w}\|_{\ell_p}^p, \tag{6}$$

where $1 < p \le 2$ and the ℓ_p-norm on $\mathbb{R}^d$ is defined as $\|\mathbf{w}\|_{\ell_p}^p = \sum_{j=1}^d |w_j|^p$. For any $1 < p' \le p \le 2$, there always holds $\|\mathbf{w}\|_{\ell_p} \le \|\mathbf{w}\|_{\ell_{p'}}$, which means that any learning method with the $\ell_{p'}$-regularizer imposes heavier penalties for the parameters than the one with the ℓ_p-regularizer. Specially when $p \to 1$, the corresponding ℓ_p-regularized algorithm tends to generate so called soft sparse solutions [Koltchinskii, 2009]. In contrast to this, the commonly-used ℓ_2-regularization tends to generate smooth but non-sparse solutions.

Note that we do not require that the minimizer of (6) is unique, catering to non-convex problems. The following lemma tells us that any global minimizer of (6) can be upper bounded by a quantity that is inversely proportional to λ.

This simple conclusion is useful in subsequent sections for designing a constrained SGD and our theoretical analysis with respect to the function $\|\mathbf{w}\|_{\ell_p}^p$.

Lemma 1. *Assume that $\ell(y, \sigma(0)) \leq B$ with some $B > 0$. For any $\lambda > 0$, any global minimizer $\widehat{\mathbf{w}}$ of (6) satisfies $\|\widehat{\mathbf{w}}\|_{\ell_p}^p \leq B/\lambda$, and furthermore $\|\widehat{\mathbf{w}}\|_{\ell_2} \leq \big(B/\lambda\big)^{1/p}$.*

Proof of Lemma 1. Since $\widehat{\mathbf{w}}$ is a global minimizer of the objective function in (6), this follows that

$$\lambda\|\widehat{\mathbf{w}}\|_{\ell_p}^p \leq \frac{1}{n}\sum\nolimits_{i=1}^{n} \ell\big(y_i, f(\mathbf{x}_i, \mathbf{0})\big) + \lambda\|\mathbf{0}\|_{\ell_p}^p \leq B.$$

Moreover, for any $1 < p \leq 2$, it is known that $\|\mathbf{w}\|_{\ell_2} \leq \|\mathbf{w}\|_{\ell_p}$ for any $\mathbf{w} \in \mathbb{R}^d$. This completes the proof of the lemma. □

Lemma 1 implies that the empirical solutions of (6) are in an ℓ_p-ball of certain radius which depends on the regularized parameter λ. This shows that it suffices to analyze statistical behaviors of the given estimator projected into this ball.

3 Regularized Stochastic Algorithm

In order to effectively solve non-convex problems with massive data, practical algorithms for machine learning are increasingly constrained to spend less time and use less memory, and can also escape from saddle points that often appear in non-convex problems and tend to converge to a good stationary point. Stochastic gradient descent (SGD) is perhaps the simplest and most well studied algorithm that enjoys these advantages. The merits of SGD for large scale learning and the associated computation versus statistics tradeoffs is discussed in detail by the seminal work of [Bottou and Bousquet, 2007].

A standard assumption to analyze SGD in the literature is that the derivative of the objective function is Lipschitz smooth [Verma and Zhang, 2019; Hardt *et al.*, 2016], however, the ℓ_p-regularized learning does not meet such condition. To address this issue, we propose a new SGD for (6) with an inexact proximal operator, and then develop a novel theoretical analysis for an upper bound of uniform stability [Bousquet and Elisseeff, 2002], which is an algorithm-dependent sensitivity-based measurement used for characterizing generalization performance in learning theory.

Given that a positive pair (p, q) satisfies the equality $1/p + 1/q = 1$, then the norms $\|\mathbf{w}\|_p$ and $\|\mathbf{w}\|_q$ are dual to each other. Moreover, the pair of functions $(1/2)\|\mathbf{w}\|_p^2$ and $(1/2)\|\mathbf{w}\|_q^2$ are conjugate functions of each other. As a consequence, their gradient mappings are a pair of inverse mapping. Formally, let $p \in (1, 2]$ and $q = p/(p-1)$, and define the mapping $\Phi : \mathrm{E} \to \mathrm{E}^*$ with

$$\Phi_j(\mathbf{w}) = \nabla_j\big(\frac{1}{2}\|\mathbf{w}\|_p^2\big) = \frac{\mathrm{sgn}(w_j)|w_j|^{p-1}}{\|\mathbf{w}\|_p^{p-2}},\ j = 1, 2, ..., d,$$

and the inverse mapping $\Phi^{-1} : \mathrm{E}^* \to \mathrm{E}$ with

$$\Phi_j^{-1}(\mathbf{v}) = \nabla_j\big(\frac{1}{2}\|\mathbf{v}\|_q^2\big) = \frac{\mathrm{sgn}(v_j)|v_j|^{q-1}}{\|\mathbf{v}\|_q^{q-2}},\ j = 1, 2, ..., d.$$

The above conjugate property on ℓ_p-space and ℓ_q-space is very useful for bounding uniform stability without the help of strong smoothness, while the latter is a standard assumption in optimization.

3.1 SGD with Inexact Proximal Operator

We write $L_i(\mathbf{w}) := \ell(y_i, f(\mathbf{x}_i, \mathbf{w}))$ for notational simplicity, and define a local quadratic approximation of L_i at point $\mathbf{w}_{D,t}$ as

$$P_{i,r_i}(\mathbf{w}, \mathbf{w}_{D,t}) := L_i(\mathbf{w}_{D,t}) + \langle \mathbf{w} - \mathbf{w}_{D,t}, \nabla L_i(\mathbf{w}_{D,t})\rangle_2 + r_i\|\mathbf{w} - \mathbf{w}_{D,t}\|_2^2.$$

At each iteration t, let i_t be a random index sampled uniformly from $[n]$ on D. Then replacing $\frac{1}{n}\sum_{i=1}^n L_i(\mathbf{w})$ in (6) by the quadratic term $P_{i_t,r_{i_t}}(\mathbf{w}, \mathbf{w}_{D,t})$, we propose an inexact proximal method for SGD with the ℓ_p-regularizer. Precisely, we are concerned with the following iterative to update $\mathbf{w}_{D,t}$ for a regularized-based SGD,

$$\min_{\mathbf{w}} P_{i_t,r_{i_t}}(\mathbf{w}, \mathbf{w}_{D,t}) + \lambda_t\|\mathbf{w}\|_{\ell_p}^p. \tag{7}$$

In view of the boundedness of $\|\widehat{\mathbf{w}}\|_{\ell_2}$ given in Lemma 1, we adopt the projection technique to execute a constrained SGD over the empirical risk term in (7). To this end, we define the projection onto a set $\mathcal{C}$ by

$$\Pi_{\mathcal{C}}(\mathbf{v}) := \arg\min_{\mathbf{w}\in\mathcal{C}} \|\mathbf{w} - \mathbf{v}\|_2.$$

This reveals that the definition of projection is an optimization problem in itself. In our case, the set we adopt is given as

$$\mathcal{C} := \mathcal{C}_\lambda = \big\{\mathbf{w} \in \mathbb{R}^d,\ \|\mathbf{w}\|_2 \leq (B/\lambda)^{1/p}\big\}.$$

It is well known that, if $\mathcal{C} = \mathcal{B}_2(1)$, i.e., the unit ℓ_2 ball, then projection is equivalent to a normalization step

$$\Pi_{\mathcal{B}_2(1)}(\mathbf{v}) = \begin{cases} \mathbf{v}/\|\mathbf{v}\|_2 & \text{if } \|\mathbf{v}\|_2 > 1, \\ \mathbf{v} & \text{otherwise.} \end{cases}$$

Up to the terms that do not depend on $\mathbf{w}$, summing the objective function in (7) and the projection formulate our proposed algorithm as follows

$$\mathbf{v}_{D,i_t} = \Pi_{\mathcal{C}_\lambda}\big(\mathbf{w}_{D,t} - (\eta\nabla L_{i_t}(\mathbf{w}_{D,t})\big), \tag{8}$$

$$\mathbf{w}_{D,t+1} = \arg\min_{\mathbf{w}} \big\{\frac{1}{2}\big\|\mathbf{w} - \mathbf{v}_{D,i_t}\big\|_2^2 + \lambda_t\|\mathbf{w}\|_{\ell_p}^p\big\}, \tag{9}$$

where $\eta > 0$ is the learning rate that depends on r_{i_t}, and the λ_t may depend on λ and r_{i_t}.

Remark 1. *The update rule in (9) is seen as a contraction of conventional SGD, see Lemma 2 below for details. We will obtain analytical solutions of (9) for some specific p (e.g. $p = 1, 2$). Although there is no analytical solutions for general $1 < p \leq 2$, the objective function in (9) is strongly convex over bounded domains and thus a global convergence can be guaranteed. For the ease of notation, we still denote by $\mathbf{w}_{D,t+1}$ the realized numerical solution with ignoring the inner optimization error.*

Lemma 2. *For $1 < p \leq 2$ and a vector $\mathbf{v} \in \mathbb{R}^d$ is given, we define*

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \big\{\frac{1}{2}\|\mathbf{w} - \mathbf{v}\|_2^2 + \lambda\|\mathbf{w}\|_{\ell_p}^p\big\} := \mathit{Pro}_{\lambda,p}(\mathbf{v}). \tag{10}$$

Then, we conclude that

$$|w_j^*| \leq \min\{|v_j|, (|v_j|/(\lambda p))^{1/(p-1)}\}, \quad \forall\, j = 1, 2..., d.$$

We defer the proof of Lemma 2 to the Appendix. Applying Lemma 2 and the projection onto $\mathcal{C}_\lambda$ in (8), we have the following inequality

$$\|\mathbf{w}_{D,t}\|_2 \leq (B/\lambda)^{1/p}, \quad \forall\, t, \lambda > 0. \tag{11}$$

Consider a function $h : \mathbb{R} \to \mathbb{R}$ defined as $h(\theta) = |\theta|^p$ $(1 < p \leq 2)$. Note that it holds

$$h'(\theta) = p.\text{sign}(\theta).|\theta|^{p-1}.$$

Obviously the first derivative of h exists and is continuous for all $\theta \in \mathbb{R}$. However, we notice that the inexact proximal operator $Pro_{\lambda,p}$ is not Lipschitz, due to the fact that $\|\cdot\|_{\ell_p}^p$ is not strongly smooth. Hence, as mentioned earlier, those conventional technique analysis under strongly smooth condition for objective functions are no longer valid in our case. Fortunately, the function h is strongly convex over bounded domains, as shown in Lemma 1, which enables us to avoid restrictive smooth assumptions by an alternative proof strategy.

4 Stability and Generalization Bounds

In this section, we provide algorithm-dependent generalization bounds via the notion of stability for ℓ_p-regularized GCN. To this end, we first introduce the notion of algorithmic stability and thereby present a generalization bound associated with the algorithmic stability.

Let $\mathcal{A}_D$ be a learning algorithm trained on dataset D, which can be viewed as a map from $D \to \mathcal{H}$. For GCN, we set $\mathcal{A}_D = f(\mathbf{x}, \mathbf{w}_D)$. The overall learning performance of $\mathcal{A}_D$ is measured by the following expected risk:

$$R(\mathcal{A}_D) := \mathbb{E}_{\mathbf{z}}\big[\ell\big(y, f(\mathbf{x}, \mathbf{w}_D)\big)\big].$$

Accordingly, the empirical risk of $\mathcal{A}_D$ with the loss ℓ is given as

$$R_n(\mathcal{A}_D) := \frac{1}{n}\sum\nolimits_{i=1}^{n} \ell\big(y_i, f(\mathbf{x}_i, \mathbf{w}_D)\big).$$

Even when the sample is fixed, $\mathcal{A}_D$ may be still a randomized algorithm due to the randomness of algorithm procedure (e.g. SGD). In this context, we define the expected generalization error as

$$E_{\text{gen}} := \mathbb{E}_{\mathcal{A}}\big[R(\mathcal{A}_D) - R_n(\mathcal{A}_D)\big],$$

where the expectation $\mathbb{E}_{\mathcal{A}}$ is taken over the inherent randomness of $\mathcal{A}_D$.

For a randomized algorithm, to introduce the notation of its uniform stability, we need to define two datasets as follows. Given the training set D defined as above, we introduce two related sets in the following:

Removing the i-th data point in the set D is represented as

$$D^{\setminus i} = \{\mathbf{z}_1, \mathbf{z}_2, ...\mathbf{z}_{i-1}, \mathbf{z}_{i+1}, ..., \mathbf{z}_n\},$$

and replacing the i-th data point in D by $\mathbf{z}_i^{'}$ is represented as

$$D^i = \{\mathbf{z}_1, \mathbf{z}_2, ...\mathbf{z}_{i-1}, \mathbf{z}_i^{'}, \mathbf{z}_{i+1}..., \mathbf{z}_n\}.$$

Definition 1. A randomized learning algorithm $\mathcal{A}_D$ is β_n-uniformly stable with respect to a loss function ℓ, if it satisfies

$$\sup_{D,\mathbf{z}} \big|\mathbb{E}_{\mathcal{A}}[\ell(y, f(\mathbf{x}, \mathbf{w}_D))] - \mathbb{E}_{\mathcal{A}}[\ell(y, f(\mathbf{x}, \mathbf{w}_{D^{\setminus i}}))]\big| \leq \beta_n.$$

By the triangle inequality, the following result on another uniform stability associated with S^i holds

$$\sup_{D,\mathbf{z}} \big|\mathbb{E}_{\mathcal{A}}[\ell(y, f(\mathbf{x}, \mathbf{w}_D))] - \mathbb{E}_{\mathcal{A}}[\ell(y, f(\mathbf{x}, \mathbf{w}_{D^i}))]\big| \leq 2\beta_n.$$

Stability is property of a learning algorithm, roughly speaking, if two training samples are close to each other, a stable algorithm will generate close output results. There are many variants of stability, such as hypothesis stability [Kearns and Ron, 1999], sample average stability [Shalev-Shwartz *et al.*, 2010] and uniform stability. This paper will focus on the uniform stability, since it is closely related to other types of stability.

The following lemma shows that a randomized learning algorithm with uniform stability can guarantee meaningful generalization bound, which has been proved in [Verma and Zhang, 2019].

Lemma 3. *A uniform stable randomized algorithm $(\mathcal{A}_D, \beta_n)$ with a bounded loss function $0 \leq \ell(y, f(\mathbf{x})) \leq B$, satisfies the following generalization bound with probability at least $1 - \delta$, over the random draw of $D, \mathbf{z}$ with $\delta \in (0, 1)$,*

$$\mathbb{E}_{\mathcal{A}}\big[R(\mathcal{A}_D) - R_n(\mathcal{A}_D)\big] \leq 2\beta_n + (4n\beta_n + B)\sqrt{\frac{\log(1/\delta)}{2n}}.$$

We now give some smooth assumptions on loss function and activation function used for analyzing the stability of stochastic gradient descent. The following assumptions are very standard in optimization literature.

Assumption 1 (Smoothness for loss function and activation function)**.** *We assume that the loss function is lipschitz continuous and smooth,*

$$|\ell(y, f(\cdot)) - \ell(y, f'(\cdot))| \leq a_\ell |f(\cdot) - f'(\cdot)|. \quad \forall\, f, f' \in \mathcal{H}$$
$$|\ell'(y, f(\cdot)) - \ell'(y, f'(\cdot))| \leq b_\ell |f(\cdot) - f'(\cdot)|,$$

where a_ℓ and b_ℓ are two positive constants. Similarly, the activation function also satisfies

$$|\sigma(x) - \sigma(y)| \leq a_\sigma |x - y|,\ |\sigma'(x) - \sigma'(y)| \leq b_\sigma |x - y| \quad \text{and } \ell(y, f(\mathbf{x})) \leq B,\ \forall\, x, y \in \mathbb{R},$$

where a_σ, b_σ and B are positive constants as well.

We now present an explicit stability bound for GCN with ℓ_p-regularizer via stochastic gradient method.

Theorem 1. *Suppose that the loss and activation functions are Lipschitz-continuous and smooth functions (Assumption 1). Then a single layer GCN model, trained by the proposed SGD given in* (8)-(9) *for iteration T, is β_n-uniformly stable, precisely*

$$\beta_n \leq a_\ell^2 a_\sigma^2 \lambda_G^{max} \frac{\eta C_{p,\lambda} \mathbf{g}_e}{n} \sum_{t=1}^{T} \Big(C_{p,\lambda}\big(1 + (a_\sigma^2 + a_\ell)\eta \mathbf{g}_e^2\big)\Big)^{t-1},$$

where $C_{p,\lambda} := \frac{28}{p(p-1)\lambda_t}\big(B/\lambda\big)^{(3-p)/p}$ *and* $\mathbf{g}_e := \sup_{\mathbf{x}} \big\| \sum_{j \in N_{\mathbf{x}}} e_{\cdot j} \mathbf{x}_j \big\|_2$.

The proof procedure of Theorem 1 will be given in Appendix. The key step of our proof is that in such a scenario, the error caused by the difference in the nonconvex empirical risk of GCN grows polynomially with the number of iterations.

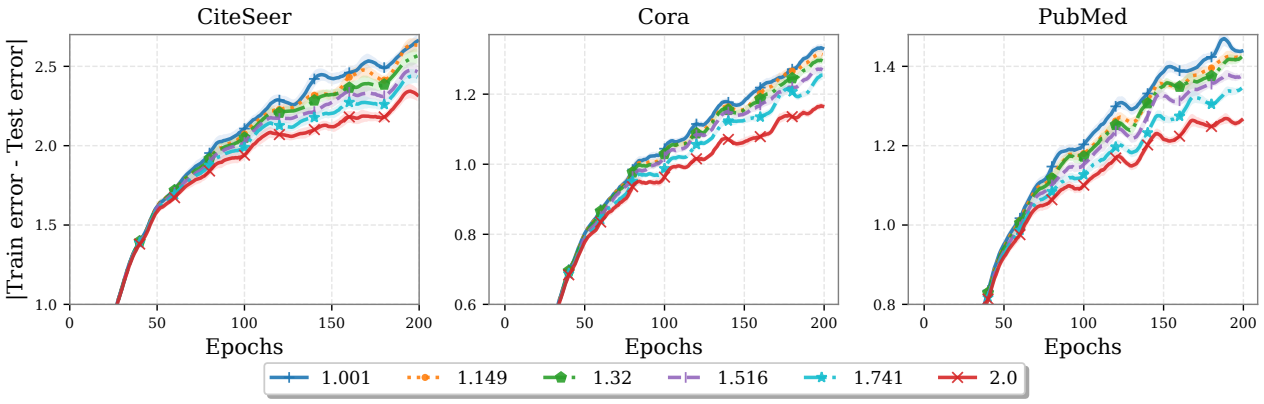

Figure 1: **Generalization Gaps as a function of the number of epochs under various p on three datasets.** We observe that the miniature p achieves weaker generalization gap and thus worse than the significant p.

Remark 2. *Theorem 1 provides the uniform stability for the last iterate of SGD for ℓ_p-regularized GCN. It is worth mentioning that the upper bound of this stability depends on the graph structure (i.e. $\mathbf{g}_e$ and λ_G^{max}) and the regularized hyper-parameter p, as well as the sample size and the learning rate in SGD.*

Remark 3. *More precisely, the result in Theorem 1 shows that the stability bound decreases inversely with the scale of p. It increases as the graph structured parameter λ_G^{max} increases.*

Substituting Lemma 3 into Theorem 1 above, we obtain the generalization bounds with uniform stability for a single layer GCN with ℓ_p-regularizer.

Theorem 2. *Under the same conditions as Theorem 1. with a high probability, the following generalization bound holds*

$$\mathbb{E}_{\mathcal{A}}\big[R(\mathcal{A}_D) - R_n(\mathcal{A}_D)\big] = \mathcal{O}\Big(\lambda_G^{max}\frac{\eta C_{p,\lambda}\mathbf{g}_e}{\sqrt{n}}\sum\nolimits_{t=1}^{T}\big(C_{p,\lambda}\big(1+(a_\sigma^2+a_\ell)\eta\mathbf{g}_e^2\big)\big)^{t-1}\Big).$$

Remark 4. *Based on the result of Theorem 2, we conclude that ℓ_p-regularization for $1 < p \leq 2$ generalizes. Note that, when $p \to 1$, the stability bound breaks due to the sparsity of ℓ_1-regularization. The smaller p becomes, the greater the stability parameter β_n is, but at the same time the obtained parameter $\widehat{\mathbf{w}}$ in* (6) *tends to be sparse, shown in [Koltchinskii, 2009]. These properties are also verified in our experimental evaluation.*

Remark 5. *Although a small learning rate means a smaller generalization gap, the parameter range in which this SGD searches will be very small, resulting in a larger training error. Such conclusion is also applicable to various SGD for general models.*

5 Experiments

In this section, we conduct experiments to validate our theoretical findings. In particular, we show that there exists a stability-sparsity trade-off with varying p, and the uniform stability of our GCN depends on the largest absolute eigenvalue of its graph filter. To do it, we first introduce the experimental settings. Then we assess the uniform stability of GCN with semi-supervised learning tasks under varying p. Finally, we evaluate the effect of different graph filters on the GCN stability bounds.

5.1 Experimental Setup

Datasets. We conduct experiments on three citation network datasets: Citeseer, Cora, and Pubmed [Sen *et al.*, 2008]. In every dataset, each document is represented as spare bag-of-words feature vectors. The relationship between documents consists of a list of citation links, which can be treated as undirected edges and help construct the adjacency matrix. These documents can be divided into different classes and have the corresponding class label.

Baselines. We implement several empirical experiments with a representative GCN model [Kipf and Welling, 2016a]. For all datasets, we use 2-layer neural networks with 16 hidden units. In all cases, we evaluate the difference between the learned weight parameters and the generalization gap of two GCN models trained on datasets D and D^i, respectively. Specifically, we generate D^i by choosing a random data point in the training set D and altering it with a different random point. We also record the training and testing errors gap and the parameter distance $\sqrt{\|\widehat{\mathbf{w}} - \widehat{\mathbf{w}}'\|^2/(\|\widehat{\mathbf{w}}\|^2 + \|\widehat{\mathbf{w}}'\|^2)}$, where $\widehat{\mathbf{w}}$ and $\widehat{\mathbf{w}}'$ are the weight parameters of the respective models per epoch.

Training settings. For each experiment, we initialize the parameters of GCN models with the same random seeds and then train all models for a maximum of 200 epochs using the proposed Inexact Proximal SGD. We repeat the experiments 10 times and report the average performance as well as the standard variance. For all methods, the hyperparameters are tuned from the following search space: (1) learning rate: $\{1, 0.5, 0.1, 0.05\}$; (2) weight decay: 0; (3) dropout rate: $\{0.3, 0.5\}$; (4) regularization parameter λ is set to 0.001.

5.2 The Effect of Varying p

Generalization gap. In this experiment, we empirically compare the effect of p on the GCN stability bounds using

Dataset	Graph Filter	$p = 1.001$	$p = 1.149$	$p = 1.320$	$p = 1.516$	$p = 1.741$	$p = 2$
Citeseer	Augmented Normalized	$\mathbf{57.44 \pm 0.87}$	54.69 ± 0.82	52.41 ± 0.73	50.72 ± 0.67	50.27 ± 0.68	50.18 ± 0.69
	Normalized	$\mathbf{57.02 \pm 1.32}$	54.47 ± 0.99	51.93 ± 0.66	50.39 ± 0.62	49.95 ± 0.62	49.88 ± 0.63
	Random Walk	$\mathbf{56.17 \pm 1.49}$	53.84 ± 1.20	51.84 ± 0.84	50.54 ± 0.78	50.16 ± 0.81	50.13 ± 0.8
	Unnormalized	$\mathbf{60.62 \pm 2.25}$	58.78 ± 2.31	57.52 ± 2.23	56.61 ± 2.25	56.32 ± 2.3	56.45 ± 2.2
Cora	Augmented Normalized	$\mathbf{56.66 \pm 3.43}$	54.46 ± 2.68	52.18 ± 2.13	50.79 ± 1.86	50.41 ± 1.77	50.27 ± 1.76
	Normalized	$\mathbf{55.77 \pm 3.68}$	53.74 ± 2.78	51.6 ± 2.10	50.22 ± 1.82	49.77 ± 1.82	49.69 ± 1.83
	Random Walk	$\mathbf{53.43 \pm 2.12}$	52.03 ± 1.94	51.08 ± 1.37	50.29 ± 1.18	50.08 ± 1.18	50.07 ± 1.17
	Unnormalized	$\mathbf{64.94 \pm 3.61}$	63.86 ± 3.5	63.31 ± 3.59	62.79 ± 3.53	62.87 ± 3.56	62.82 ± 3.5
Pubmed	Augmented Normalized	$\mathbf{54.13 \pm 2.16}$	52.61 ± 2.62	50.83 ± 1.92	49.73 ± 1.67	49.24 ± 1.69	49.06 ± 1.63
	Normalized	$\mathbf{53.27 \pm 3.40}$	52.48 ± 2.35	51.28 ± 2.09	50.29 ± 1.96	49.56 ± 1.78	49.30 ± 1.55
	Random Walk	$\mathbf{54.72 \pm 3.57}$	53.07 ± 3.20	51.82 ± 2.94	51.03 ± 2.84	50.85 ± 2.8	50.77 ± 2.76
	Unnormalized	$\mathbf{74.54 \pm 6.26}$	74.10 ± 6.40	73.29 ± 6.3	73.25 ± 6.57	72.91 ± 6.12	73.09 ± 5.96

Table 1: The sparsity (%) of obtained parameter $\widehat{\mathbf{w}}$ under different p and normalization steps on three datasets.

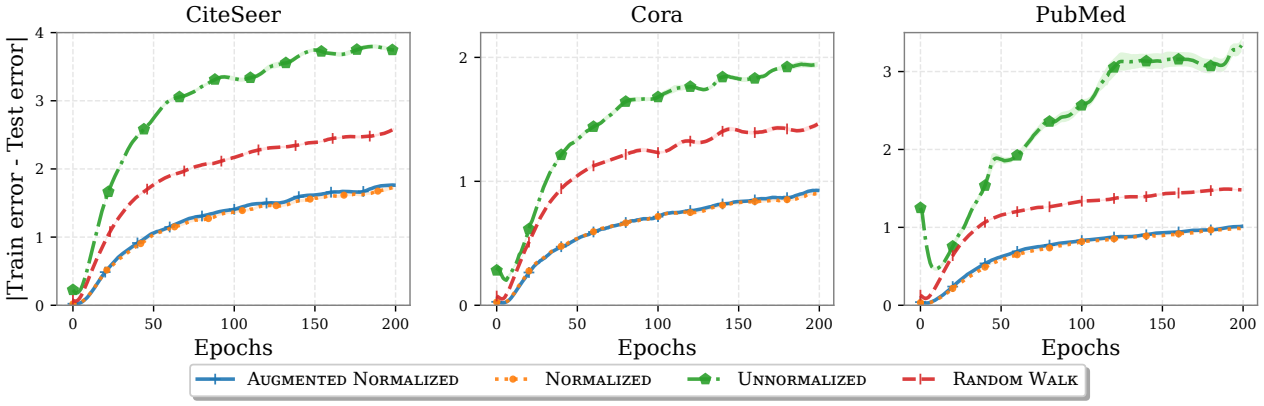

Figure 2: **Generalization Gaps under different normalization steps on three datasets.** We observe that the normalized filters yield significantly lower generalization gaps than the unnormalized and random walk filters.

different $p \in \{1.001, 1.149, 1.32, 1.516, 1.741, 2\}$. We quantitatively measure the generalization gap defined as the absolute difference between the training and testing errors and the difference between learned weights parameters of GCN models trained on D and D', respectively. It can be observed that in Figure 1, these empirical observations are in line with our stability bounds (see also results in Figure 3 in Appendix).

Sparsity. Besides the convergence, we have a particular concern about the sparsity of the solutions. The sparsity ratios for l_p-based method are summarized in Table 1. Observe that ℓ_p-regularized learning with $p \to 1$ identifies most of the sparsity pattern but behaves much worse in generalization.

5.3 The Effect of Graph Filters

Different normalizations steps. In this experiment, we mainly consider the implication of our results in following designing graph convolution filters: (1) Unnormalized Graph Filters: $g(\mathbf{L}) = \mathbf{A} + \mathbf{I}$; (2) Normalized Graph Filters: $g(\mathbf{L}) = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} + \mathbf{I}$; (3) Random Walk Graph Filters: $g(\mathbf{L}) = \mathbf{D}^{-1}\mathbf{A} + \mathbf{I}$; (4) Augmented Normalized Graph Filters: $g(\mathbf{L}) = (\mathbf{D} + \mathbf{I})^{-1/2}(\mathbf{A} + \mathbf{I})(\mathbf{D} + \mathbf{I})^{-1/2}$.

In this experiment, we quantitatively measure the generalization gap and parameter distance per epoch. From Figure 2, it is clear that the Unnormalized Graph Filters and Random Walk Graph Filters show a significantly higher generalization gap than the normalized ones. The results hold consistently across the three datasets. Hence, these empirical results are also consistent with our generalization error bound. We note that the generalization gap and parameter distance become constant after a certain number of iterations. More results can be found in the supplementary material.

6 Conclusion

In this paper, we present an explicit theoretical understanding of stochastic learning for GCN with the l_p regularizer and analyze the stability of our regularized stochastic algorithm. In particular, our derived bounds show that the uniform stability of our GCN depends on the largest absolute eigenvalue of its graph filter, and there exists a generalization-sparsity tradeoff with varying p. It is worth noting that previous generalization analysis based on stability notation assumed that objectiveness is a second derivative function, which is no longer applicable to our l_p-regularized learning scheme. To address this issue, we propose a new proximal SGD algorithm for GCNs with an inexact operator, which exhibits comparable empirical performances.

Acknowledgements

Lv's research is supported by the "QingLan" Project of Jiangsu Universities. Ming Li would like to thank the support from the National Natural Science Foundation of China (No. U21A20473, No. 62172370) and Zhejiang Provincial Natural Science Foundation (No. LY22F020004).

Contribution Statement

Linsen We and Ming Li contributed equally to this work.

References

[Barrett *et al.*, 2018] David Barrett, Felix Hill, Adam Santoro, Ari Morcos, and Timothy Lillicrap. Measuring abstract reasoning in neural networks. In *International Conference on Machine Learning*, pages 511–520. PMLR, 2018.

[Battaglia *et al.*, 2016] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems*, pages 4502–4510, 2016.

[Bottou and Bousquet, 2007] Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems*, pages 161–168, 2007.

[Bousquet and Elisseeff, 2002] Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

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

[Defferrard *et al.*, 2016] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, volume 29, pages 3844–3852, 2016.

[Duvenaud *et al.*, 2015] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems*, volume 28, pages 2224–2232, 2015.

[Garg *et al.*, 2020] Vikas Garg, Stefanie Jegelka, and Tommi Jaakkola. Generalization and representational limits of graph neural networks. In *International Conference on Machine Learning*, pages 3419–3430. PMLR, 2020.

[Gilmer *et al.*, 2017] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pages 1263–1272. PMLR, 2017.

[Hamilton *et al.*, 2017] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035, 2017.

[Hardt *et al.*, 2016] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 1225–1234. PMLR, 2016.

[Jin *et al.*, 2018] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International Conference on Machine Learning*, pages 2323–2332. PMLR, 2018.

[Kearns and Ron, 1999] Michael Kearns and Dana Ron. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. *Neural Computation*, 11(6):1427–1453, 1999.

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

[Kipf and Welling, 2016b] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[Koltchinskii, 2009] Vladimir Koltchinskii. Sparsity in penalized empirical risk minimization. In *Annales de l'IHP Probabilités et statistiques*, volume 45, pages 7–57, 2009.

[Liao *et al.*, 2020] Renjie Liao, Raquel Urtasun, and Richard Zemel. A pac-bayesian approach to generalization bounds for graph neural networks. In *International Conference on Learning Representations*, 2020.

[Liu *et al.*, 2021] Xiaorui Liu, Wei Jin, Yao Ma, Yaxin Li, Hua Liu, Yiqi Wang, Ming Yan, and Jiliang Tang. Elastic graph neural networks. In *International Conference on Machine Learning*, pages 6837–6849. PMLR, 2021.

[Ma *et al.*, 2021] Yao Ma, Xiaorui Liu, Tong Zhao, Yozen Liu, Jiliang Tang, and Neil Shah. A unified view on graph neural networks as graph signal denoising. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1202–1211, 2021.

[Nie *et al.*, 2011] Feiping Nie, Hua Wang, Heng Huang, and Chris Ding. Unsupervised and semi-supervised learning via 1-norm graph. In *2011 International Conference on Computer Vision*, pages 2268–2273. IEEE, 2011.

[Oono and Suzuki, 2020] Kenta Oono and Taiji Suzuki. Optimization and generalization analysis of transduction through gradient boosting and application to multi-scale graph neural networks. *Advances in Neural Information Processing Systems*, 33:18917–18930, 2020.

[Schlichtkrull *et al.*, 2018] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer, 2018.

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

[Shalev-Shwartz *et al.*, 2010] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *The Journal of Machine Learning Research*, 11:2635–2670, 2010.

[Tibshirani, 2014] Ryan J Tibshirani. Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, 42(1):285–323, 2014.

[Veličković *et al.*, 2017] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[Verma and Zhang, 2019] Saurabh Verma and Zhi-Li Zhang. Stability and generalization of graph convolutional neural networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1539–1548, 2019.

[Wang *et al.*, 2015] Yu-Xiang Wang, James Sharpnack, Alex Smola, and Ryan Tibshirani. Trend filtering on graphs. In *Artificial Intelligence and Statistics*, pages 1042–1050. PMLR, 2015.

[Wibisono *et al.*, 2009] Andre Wibisono, Lorenzo Rosasco, and Tomaso Poggio. Sufficient conditions for uniform stability of regularization algorithms. *Computer Science and Artificial Intelligence Laboratory Technical Report, MIT-CSAIL-TR-2009-060*, 2009.

[Xu *et al.*, 2011] Huan Xu, Constantine Caramanis, and Shie Mannor. Sparse algorithms are not stable: A no-free-lunch theorem. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 34(1):187–193, 2011.

[Xu *et al.*, 2018] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[Yun *et al.*, 2019] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in Neural Information Processing Systems*, 32:11983–11993, 2019.

[Zhang *et al.*, 2020] Yuyu Zhang, Xinshi Chen, Yuan Yang, Arun Ramamurthy, Bo Li, Yuan Qi, and Le Song. Efficient probabilistic logic reasoning with graph neural networks. *arXiv preprint arXiv:2001.11850*, 2020.