On Provable Benefits of Depth in Training Graph Convolutional Networks

Weilin Cong
Penn State
wxc272@psu.edu

Morteza Ramezani
Penn State
morteza@cse.psu.edu

Mehrdad Mahdavi
Penn State
mzm616@psu.edu

Abstract

Graph Convolutional Networks (GCNs) are known to suffer from performance degradation as the number of layers increases, which is usually attributed to over-smoothing. Despite the apparent consensus, we observe that there exists a discrepancy between the theoretical understanding of over-smoothing and the practical capabilities of GCNs. Specifically, we argue that over-smoothing does not necessarily happen in practice, a deeper model is provably expressive, can converge to global optimum with linear convergence rate, and achieve very high training accuracy as long as properly trained. Despite being capable of achieving high training accuracy, empirical results show that the deeper models generalize poorly on the testing stage and existing theoretical understanding of such behavior remains elusive. To achieve better understanding, we carefully analyze the *generalization capability* of GCNs, and show that the training strategies to achieve high training accuracy significantly deteriorate the *generalization capability* of GCNs. Motivated by these findings, we propose a decoupled structure for GCNs that detaches weight matrices from feature propagation to preserve the expressive power and ensure good generalization performance. We conduct empirical evaluations on various synthetic and real-world datasets to validate the correctness of our theory.

1 Introduction

In recent years, Graph Convolutional Networks (GCNs) have achieved state-of-the-art performance in dealing with graph-structured applications, including social networks [27, 20, 50, 11, 43], traffic prediction [9, 44, 33, 30], knowledge graphs [51, 52, 42], drug reaction [13, 15] and recommendation system [2, 59]. Despite the success of GCNs, applying a shallow GCN model that only uses the information of a very limited neighborhood on a large sparse graph has shown to be not effective [22, 6, 19, 8, 45]. As a result, a deeper GCN model would be desirable to reach and aggregate information from farther neighbors. The inefficiency of shallow GCNs is exacerbated even further when the labeled nodes compared to graph size is negligible, as a shallow GCN cannot sufficiently propagate the label information to the entire graph with only a few available labels [34].

Although a deeper GCN is preferred to perceive more graph structure information, unlike traditional deep neural networks, it has been pointed out that deeper GCNs potentially suffer from over-smoothing [34, 40, 25, 4, 57], vanishing/exploding gradients [32], over-squashing [1], and training difficulties [61, 37], which significantly affect the performance of GCNs as the depth increases. Among these, the most widely accepted reason is "over-smoothing", which is referred to as a phenomenon due to applying multiple graph convolutions such that all node embeddings converge to a single subspace (or vector) and leads to indistinguishable node representations.

The conventional wisdom is that adding to the number of layers causes over-smoothing, which impairs the *expressiveness power* of GCNs and consequently leads to a poor *training accuracy*. However, we observe that there exists a discrepancy between theoretical understanding of their inherent capabilities

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

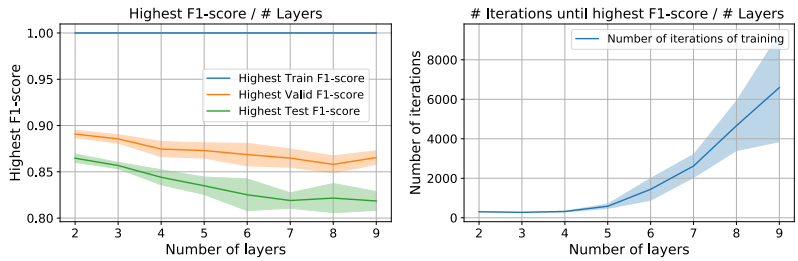

Figure 1: Comparison of F1-score for GCN with different depth on *Cora* dataset, where deeper models can achieve high training accuracy, but complicate the training by requiring more iterations to converge and suffer from poor generalization.

and practical performances. According to the definition of over-smoothing that the node representation becomes indistinguishable as GCNs goes deeper, the classifier has difficulty assigning the correct label for each node if over-smoothing happens. As a result, the training accuracy is expected to be decreasing as the number of layers increases. However, as shown in Figure 1, GCNs are capable of achieving high *training accuracy* regardless of the number of layers. But, as it can be observed, deeper GCNs require more training iterations to reach a high training accuracy, and its *generalization performance* on evaluation set decreases as the number of layers increases. This observation suggests that the performance degradation is likely due to inappropriate training rather than the low expressive power caused by over-smoothing. Otherwise, a low expressiveness model cannot achieve almost perfect training accuracy simply by proper training tricks alone.[1] Indeed, recent years significant advances have been witnessed on tweaking the model architecture to overcome the training difficulties in deeper GCN models and achieve good generalization performance [37, 6, 32, 61].

Contributions. Motivated by aforementioned observation, i.e., still achieving high training accuracy when trained properly but poor generalization performance, we aim at answering two fundamental questions in this paper:

Q1: *Does increasing depth really impair the expressiveness power of GCNs?*
In Section 4, we argue that there exists a discrepancy between over-smoothing based theoretical results and the practical capabilities of deep GCN models, demonstrating that over-smoothing is not the key factor that leads to the performance degradation in deeper GCNs. In particular, we mathematically show that over-smoothing [40, 25, 34, 4] is mainly an artifact of theoretical analysis and simplifications made in analysis. Indeed, by characterizing the representational capacity of GCNs via Weisfeiler-Lehman (WL) graph isomorphism test [38, 54], we show that deeper GCN model is at least as expressive as the shallow GCN model, the deeper GCN models can distinguish nodes with a different neighborhood that the shallow GCN cannot distinguish, as long as the GCNs are properly trained. Besides, we theoretically show that more training iterations is sufficient (but *not necessary* due to the assumptions made in our theoretical analysis) for a deeper model to achieve the same training error as the shallow ones, which further suggests the poor training error in deep GCN training is most likely due to inappropriate training.

Q2: *If expressive, why then deep GCNs generalize poorly?*
In Section 5, in order to understand the performance degradation phenomenon in deep GCNs during the evaluation phase, we give a novel generalization analysis on GCNs and its variants (e.g., ResGCN, APPNP, and GCNII) under the semi-supervised setting for the node classification task. We show that the generalization gap of GCNs is governed by the number of training iterations, largest node degree, the largest singular value of weight matrices, and the number of layers. In particular, our result suggests that a deeper GCN model requires more iterations of training and optimization tricks to converge (e.g., adding skip-connections), which leads to a poor generalization. More interestingly, our generalization analysis shows that most of the so-called methods to solve over-smoothing [46, 60, 6, 28] can greatly improve the generalization ability of the model, therefore results in a deeper model.

[1] The results in Figure 1 can be reproduced by removing both the dropout and weight decay operation. These two augmentations are designed to improve the generalization ability (i.e., validation/testing accuracy) of the model but might hurt the training accuracy, because of the randomness and also an extra regularization term on the model parameters which are introduced during training. A simple experiment using DGL can be found here.

The aforementioned findings naturally lead to the algorithmic contribution of this paper. In Section 6, we present a novel framework, ***D**ecoupled **GCN*** (DGCN), that is capable of training deeper GCNs and can significantly improve the generalization performance. The main idea is to isolate the expressive power from generalization ability by decoupling the weight parameters from feature propagation. In Section 7, we conduct experiments on the synthetic and real-world datasets to validate the correctness of the theoretical analysis and the advantages of DGCN over baseline methods.

2 Related works

Expressivity of GCNs. Existing results on expressive power of GCNs are mixed, [38, 36, 7, 5] argue that deeper model has higher expressive power but [40, 25, 24] have completely opposite result. On the one hand, [38] shows a deeper GCN is as least as powerful as the shallow one in terms of distinguishing non-isomorphic graphs. [36] shows that deep and wide GCNs is Turing universal, however, GCNs lose power when their depth and width are restricted. [7] and [5] measure the expressive power of GCNs via its subgraph counting capability and attribute walks, and both show the expressive power of GCNs grows exponentially with the GCN depth. On the other hand, [24] studies the infinity wide GCNs and shows that the covariance matrix of its outputs converges to a constant matrix at an exponential rate. [40, 25] characterize the expressive power using the distance between node embeddings to a node feature agnostic subspace and show the distance is decreasing as the number of layers increases. Details are deferred to Section 4 and Appendix B. Such contradictory results motivate us to rethink the role of over-smoothing on expressiveness.

Generalization analysis of GCNs. In recent years, many papers are working on the generalization of GCNs using uniform stability [49, 62], Neural Tangent Kernel [14], VC-dimension [48], Rademacher complexity [17, 41], algorithm alignment [55, 56] and PAC-Bayesian [35]. Existing works only focus on a specific GCN structure, which cannot be used to understand the impact of GCN structures on its generalization ability. The most closely related to ours is [49], where they analyze the stability of the single-layer GCN model, and show that the stability of GCN depends on the largest absolute eigenvalue of its Laplacian matrix. However, their result is under the inductive learning setting and extending the results to the multi-layer GCNs with different structures is non-trivial.

Literature with similar observations. Most recently, several works have similar observations on the over-smoothing issue to ours. [61] argues that the main factors to performance degradation are vanishing gradient, training instability, and over-fitting, rather than over-smoothing, and proposes a node embedding normalization heuristic to alleviate the aforementioned issues. [37] argues that the performance degradation is mainly due to the training difficulty, and proposes a different graph Laplacian formulation, weight parameter initialization, and skip-connections to improve the training difficulty. [58] argues that deep GCNs can learn to overcome the over-smoothing issue during training, and the key factor of performance degradation is over-fitting, and proposes a node embedding normalization method to help deep GCNs overcome the over-smoothing issue. [29] improves the generalization ability of GCNs by using adversarial training and results in a consistent improvement than all GCNs baseline models without adversarial training. All aforementioned literature only gives heuristic explanations based on the empirical results, and do not provide theoretical arguments.

3 Preliminaries

Notations and setup. We consider the semi-supervised node classification problem, where a self-connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $N = |\mathcal{V}|$ nodes is given in which each node $i \in \mathcal{V}$ is associated with a feature vector $\mathbf{x}_i \in \mathbb{R}^{d_0}$, and only a subset of nodes $\mathcal{V}_{\text{train}} \subset \mathcal{V}$ are labeled, i.e., $y_i \in \{1, \ldots, |\mathcal{C}|\}$ for each $i \in \mathcal{V}_{\text{train}}$ and $\mathcal{C}$ is the set of all candidate classes. Let $\mathbf{A} \in \mathbb{R}^{N \times N}$ denote the adjacency matrix and $\mathbf{D} \in \mathbb{R}^{N \times N}$ denote the corresponding degree matrix with $D_{i,i} = \deg(i)$ and $D_{i,j} = 0$ if $i \neq j$. Then, the propagation matrix (using the Laplacian matrix defined in [27]) is computed as $\mathbf{P} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$. Our goal is to learn a GCN model using the node features for all nodes $\{\mathbf{x}_i\}_{i \in \mathcal{V}}$ and node labels for the training set nodes $\{y_i\}_{i \in \mathcal{V}_{\text{train}}}$, and expect it generalizes well on the unlabeled node set $\mathcal{V}_{\text{test}} = \mathcal{V} \setminus \mathcal{V}_{\text{train}}$.

GCN architectures. In this paper, we consider the following architectures for training GCNs:

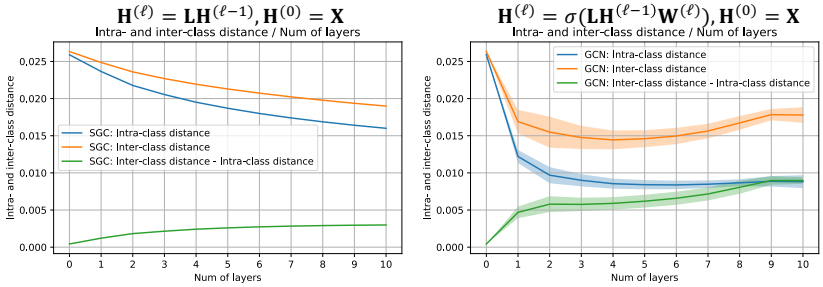

Figure 2: Comparison of intra- and inter-class normalized node embeddings $\mathbf{H}^{(\ell)}/\|\mathbf{H}^{(\ell)}\|_{\mathrm{F}}$ pairwise distance on *Cora* dataset. See Appendix A for more evidences.

- ***Vanilla GCN*** [27] computes node embeddings by $\mathbf{H}^{(\ell)} = \sigma(\mathbf{P}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)})$, where $\mathbf{H}^{(\ell)} = \{\mathbf{h}_i^{(\ell)}\}_{i=1}^N$ is the ℓth layer node embedding matrix, $\mathbf{h}_i^{(\ell)} \in \mathbb{R}^{d_\ell}$ is the embedding of ith node, $\mathbf{W}^{(\ell)} \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ is the ℓth layer weight matrix, and $\sigma(\cdot)$ is the ReLU activation.
- ***ResGCN*** [32] solves the vanishing gradient issue by adding skip-connections between adjacency layers. More specifically, ResGCN computes node embeddings by $\mathbf{H}^{(\ell)} = \sigma(\mathbf{P}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)}) + \mathbf{H}^{(\ell-1)}$, where node embeddings of the previous layer is added to the output of the current layer to facilitate the training of deeper GCN models.
- ***APPNP*** [28] adds skip-connections from the input layer to each hidden layer to preserve the feature information. APPNP computes node embeddings by $\mathbf{H}^{(\ell)} = \alpha_\ell \mathbf{P}\mathbf{H}^{(\ell-1)} + (1-\alpha_\ell)\mathbf{H}^{(0)}\mathbf{W}$, where $\alpha_\ell \in [0,1]$ balances the amount of information preserved at each layer. By decoupling feature transformation and propagation, APPNP can aggregate information from multi-hop neighbors without significantly increasing the computation complexity.
- ***GCNII*** [6] improves the capacity of APPNP by adding non-linearty and weight matrix at each individual layer. GCNII computes node embeddings by $\mathbf{H}^{(\ell)} = \sigma\big((\alpha_\ell \mathbf{P}\mathbf{H}^{(\ell-1)} + (1-\alpha_\ell)\mathbf{H}^{(0)})\bar{\mathbf{W}}^{(\ell)}\big)$, where $\bar{\mathbf{W}}^{(\ell)} = \beta_\ell \mathbf{W}^{(\ell)} + (1-\beta_\ell)\mathbf{I}$, constant α_ℓ same as APPNP, and constant $\beta_\ell \in [0,1]$ restricts the power of ℓth layer parameters.

4 On true expressiveness and optimization landscape of deep GCNs

Empirical validation of over-smoothing. In this section, we aim at answering the following fundamental question: *"Does over-smoothing really cause the performance degradation in deeper GCNs?"* As first defined in [34], *over-smoothing* is referred to as a phenomenon where all node embeddings converge to a single vector after applying multiple graph convolution operations to the node features. However, [34] only considers the graph convolution operation *without* non-linearity and the per-layer weight matrices. To verify whether over-smoothing exists in normal GCNs, we measure the pairwise distance between the normalized node embeddings with varying the model depth.[2] As shown in Figure 2, without the weight matrices and non-linear activation functions, the pairwise distance between node embeddings indeed decreases as the number of layers increases. However, by considering the weight matrices and non-linearity, the pairwise distances are actually increasing after a certain depth which *contradicts* the definition of over-smoothing that node embeddings become indistinguishable when the model becomes deeper. That is, graph convolution makes adjacent node embeddings get closer, then non-linearity and weight matrices help node embeddings preserve distinguishing-ability after convolution.

Most recently, [40, 25] generalize the idea of over-smoothing by taking both the non-linearity and weight matrices into considerations. More specifically, the expressive power of the ℓth layer node embeddings $\mathbf{H}^{(\ell)}$ is measured using $d_{\mathcal{M}}(\mathbf{H}^{(\ell)})$, which is defined as the distance of node embeddings to a subspace $\mathcal{M}$ that only has node degree information. Let denote λ_L as the second largest eigenvalue of graph Laplacian and λ_W as the largest singular value of weight matrices. [40, 25] show that the expressive power $d_{\mathcal{M}}(\mathbf{H}^{(\ell)})$ is bounded by $d_{\mathcal{M}}(\mathbf{H}^{(\ell)}) \leq (\lambda_W \lambda_L)^\ell \cdot d_{\mathcal{M}}(\mathbf{X})$, i.e., the expressive power of node embeddings will be exponentially decreasing or increasing as the number

[2] The pairwise distances are computed on the normalized node embeddings $\mathbf{H}^{(\ell)}/\|\mathbf{H}^{(\ell)}\|_{\mathrm{F}}^2$ to eliminate the effect of node embedding norms.

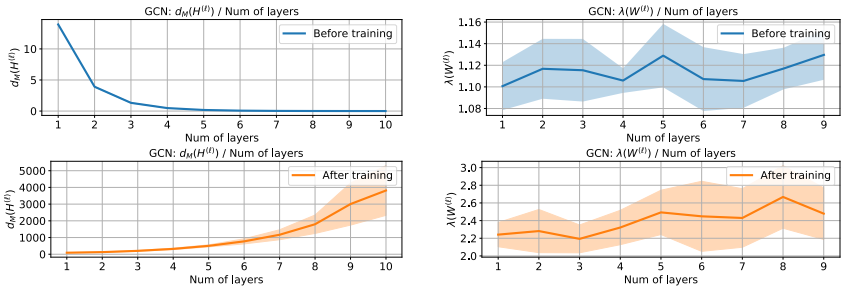

Figure 3: Compare $d_{\mathcal{M}}(\mathbf{H}^{(\ell)})$ and $\lambda_W(\mathbf{W}^{(\ell)})$ on both *trained*- (using 500 gradient update) and *untrained*-GCN models on *Cora* dataset.

of layers increases, depending on whether $\lambda_W \lambda_L < 1$ or $\lambda_W \lambda_L > 1$. They conclude that *deeper GCN exponentially loss expressive power* by assuming $\lambda_W \lambda_L < 1$. However, we argue that this assumption does not always hold. To see this, let suppose $\mathbf{W}^{(\ell)} \in \mathbb{R}^{d_{\ell-1} \times d_\ell}$ is initialized by uniform distribution $\mathcal{N}(0, \sqrt{1/d_{\ell-1}})$. By the Gordon's theorem for Gaussian matrices [10], we know its expected largest singular value is bounded by $\mathbb{E}[\lambda_W] \leq 1 + \sqrt{d_\ell / d_{\ell-1}}$, which is strictly greater than 1 and λ_W usually increases during training. The above discussion also holds for other commonly used initialization methods [18, 21]. Furthermore, since most real world graphs are sparse with λ_L close to 1, e.g., Cora has $\lambda_L = 0.9964$, Citeseer has $\lambda_L = 0.9987$, and Pubmed has $\lambda_L = 0.9905$, making assumption on $\lambda_W \lambda_L < 1$ is not realistic. As shown in Figure 3, when increasing the number of layers, we observe that the distance $d_{\mathcal{M}}(\mathbf{H}^{(\ell)})$ is decreasing on *untrained*-GCN models, however, the distance $d_{\mathcal{M}}(\mathbf{H}^{(\ell)})$ is increasing on *trained*-GCN models, which contradicts the conclusion in [40]. Due to the space limit, we defer the more empirical evidence to Appendix B. Through these findings, we cast doubt on the power of over-smoothing based analysis to provide a complete picture of why deep GCNs perform badly.

Definition 1. *Let $\mathcal{T}_i^L$ denote the L-layer computation tree of node i, which represents the structured L-hop neighbors of node i, where the children of any node j in the tree are the nodes in $\mathcal{N}(i)$.*

Exponentially growing expressieness without strong assumptions. Indeed, we argue that *deeper GCNs have stronger expressive power than the shallow GCNs.* To prove this, we employ the connection between WL test[3] [31] and GCNs. Recently, [38] shows that GCNs have the same expressiveness as the WL-test for graph isomorphism if they are appropriately trained, i.e., a properly trained L-layer GCN computes different node representations for two nodes if their L-layer *computation tree* (Definition 1) have different structure or different features on the corresponding nodes. Since L-GCN can encode any different computation tree into different representations, it is natural to characterize the expressiveness of L-GCN by the number of computation graph it can encode.

Theorem 1. *Suppose $\mathcal{T}^L$ is a computation tree with binary node features and node degree at least d. Then, by assuming the computation tree of two nodes are disjoint, the richness (i.e., the number of computation graphs a model can encode) of the output of L-GCN defined on $\mathcal{T}^L$ is at least $|L\text{-}GCN(\mathcal{T}^L)| \geq 2(d-1)^{L-1}$.*

The proof is deferred to Appendix C. The above theorem implies that the richness of L-GCN grows at least exponentially with respect to the number of layers.

Comparison of expressiveness metrics. Although distance-based expressiveness metric [40, 25] is strong than WL-based metric in the sense that node embeddings can be *distinct* but *close to each other*, the distance-based metric requires explicit assumptions on the GCN structures, weight matrices, and graph structures comparing to the WL-based metric, which has been shown that are not likely hold. On the other hand, WL-based metric has been widely used in characterizing the expressive power of GCNs in graph-level task [38, 36, 7, 5]. More details are deferred to related works (Section 2).

[3] WL test is a recursive algorithm where the label of a node depends on its own label and neighbors from the previous iterations, i.e., $c_i^{(\ell)} = \text{Hash}(c_i^{(\ell-1)}, \{c_j^{(\ell-1)} | j \in \mathcal{N}(i)\})$, where $\text{Hash}(\cdot)$ bijectively maps a set of values to a unique value that has not been used in the previous iterations. After L-iterations, the WL test will assign two nodes with a different label if the L-hop neighborhood of two nodes are non-isomorphic.

Although expressive, it is still unclear why the deeper GCN requires more training iterations to achieve small training error and reach the properly trained status. To understand this, we show in Theorem 2 that under assumptions on the width of the final layer, the deeper GCN can converge to its global optimal with linear convergence rate. More specifically, the theorem claims that if the dimension of the last layer of GCN d_L is larger than the number of data N[4], then we can guarantee the loss $\mathcal{L}(\boldsymbol{\theta}_T) \leq \varepsilon$ after $T = \mathcal{O}(1/\varepsilon)$ iterations of the gradient updates. Besides, more training iterations is sufficient (but not necessary due to the assumptions made in our theoretical analysis) for a deeper model to achieve the same training error as the shallow ones.

Theorem 2. *Let $\boldsymbol{\theta}_t = \{\mathbf{W}_t^{(\ell)} \in \mathbb{R}^{d_{\ell-1}\times d_\ell}\}_{\ell=1}^{L+1}$ be the model parameter at the t-th iteration and using square loss $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2}\|\mathbf{H}^{(L)}\mathbf{W}^{(L+1)} - \mathbf{Y}\|_{\mathrm{F}}^2$, $\mathbf{H}^{(\ell)} = \sigma(\mathbf{P}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)})$ as objective function. Then, under the condition that $d_L \geq N$ we can obtain $\mathcal{L}(\boldsymbol{\theta}_T) \leq \epsilon$ if $T \geq C(L)\log(\mathcal{L}(\boldsymbol{\theta}_0)/\epsilon)$, where ϵ is the desired error and $C(L)$ is a function of GCN depth L that grows as GCN becomes deeper.*

A formal statement of Theorem 2 and its proof are deferred to Appendix D. Besides, gradient stability also provides an alternative way of empirically understanding why deeper GCN requires more iterations: deeper neural networks are prone to exploding/vanishing gradient, which results in a very noisy gradient and requires small learning rate to stabilize the training. This issue can be significantly alleviated by adding skip-connections (Appendix E.5). When training with adaptive learning rate mechanisms, such as Adam [26][5], noisy gradient will result in a much smaller update on current model compared to a stabilized gradient, therefore more training iterations are required.

5 A different view from generalization

In the previous section, we provided evidence that a well-trained deep GCN is at least as powerful as a shallow one. However, it is still unclear why a deeper GCN has worse performance than a shallow GCN during the evaluation phase. To answer this question, we provide a different view by analyzing the impact of GCN structures on the generalization.

Transductive uniform stability. In the following, we study the generalization ability of GCNs via *transductive uniform stability* [16], where the generalization gap is defined as the difference between the training and testing errors for the random partition of a full dataset into training and testing sets. Transductive uniform stability is defined under the notation that the output of a classifier does not change much if the input is perturbed a bit, which is an extension of uniform stability [3] from the inductive to the transductive setting. The previous analysis on the uniform stability of GCNs [49] only shows the result of GCN with one graph convolutional layer under inductive learning setting, which cannot explain the effect of depth, model structure, and training data size on the generalization, and its extension to multi-layer GCNs and other GCN structures are non-trivial.

Problem setup. Let $m = |\mathcal{V}_{\text{train}}|$ and $u = |\mathcal{V}_{\text{test}}|$ denote the training and test dataset sizes, respectively. Under the transductive learning setting, we start with a fixed set of points $X_{m+u} = \{x_1, \ldots, x_{m+u}\}$. For notational convenience, we assume X_m are the first m data points and X_u are the last u data points of X_{m+u}. We randomly select a subset $X_m \subset X_{m+u}$ uniformly at random and reveal the labels Y_m for the selected subset for training, but the labels for the remaining u data points $Y_u = Y_{m+u} \setminus Y_m$ are not available during the training phase. Let $S_m = ((x_1, y_1), \ldots, (x_m, y_m))$ denotes the labeled set and $X_u = (x_{m+1}, \ldots, x_{m+u})$ denotes the unlabeled set. Our goal is to learn a model to label the remaining unlabeled set as accurately as possible.

For the analysis purpose, we assume a binary classifier is applied to the final layer node representation $f(\mathbf{h}_i^{(L)}) = \tilde{\sigma}(\mathbf{v}^\top \mathbf{h}_i^{(L)})$ with $\tilde{\sigma}(\cdot)$ denotes the sigmoid function. We predict $\hat{y}_i = 1$ if $f(\mathbf{h}_i^{(L)}) > 1/2$ and $\hat{y}_i = 0$ otherwise, with ground truth label $y_i \in \{0, 1\}$. Let denote the perturbed dataset as $S_m^{ij} \triangleq (S_m \setminus \{(x_i, y_i)\}) \cup \{(x_j, y_j)\}$ and $X_u^{ij} \triangleq (X_u \setminus \{x_j\}) \cup \{x_i\}$, which is obtained by replacing the ith example in training set S_m with the jth example from the testing set X_u. Let $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^{ij}$ denote the weight parameters trained on the original dataset (S_m, X_u) and the perturbed dataset (S_m^{ij}, X_u^{ij}),

[4]This type of over-parameterization assumptions are required and commonly used in other neural network convergence analysis to guarantee that the model parameters do not change significantly during training.

[5]In the Adam optimizer, the contribution of bias correction of moments varies exponentially over epochs completed. Although the learning rate hyper-parameter is a constant, the contribution of gradients to updated weight varies over epochs, hence adaptive. Please refer to the Chapter 8.5 of [47] for more details.

respectively. Then, we say transductive learner f is ϵ-uniformly stable if the outputs change less than ϵ when we exchange two examples from the training set and testing set, i.e., for any $S_m \subset S_{m+u}$ and any $i, j \in [m+u]$ it holds that $\sup_{i\in[m+u]} |f(\mathbf{h}_i^{(L)}) - \tilde{f}(\tilde{\mathbf{h}}_i^{(L)}))| \leq \epsilon$, where $f(\mathbf{h}_i^{(L)})$ and $\tilde{f}(\tilde{\mathbf{h}}_i^{(L)})$ denote the prediction of node i using parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^{ij}$ respectively.

To define testing error and training error in our setting, let us introduce the difference in probability between the correct and the incorrect label as $p(z, y) \triangleq y(2z-1)+(1-y)(1-2z)$ with $p(z, y) \leq 0$ if there exists a classification error. Let denote the γ-margin loss as $\Phi_\gamma(x) = \min(1, \max(0, 1-x/\gamma))$. Then, the testing error is defined as $\mathcal{R}_u(f) = \frac{1}{u}\sum_{i=m+1}^{m+u} \mathbf{1}\{p(f(\mathbf{h}_i^{(L)}), y_i) \leq 0\}$ and the training loss is defined as $\mathcal{R}_m^\gamma(f) = \frac{1}{m}\sum_{i=1}^{m} \Phi_\gamma(-p(f(\mathbf{h}_i^{(L)}), y_i))$. [6]

Theorem 3 (Transductive uniform stability bound [16])**.** *Let f be a ϵ-uniformly stable transductive learner and $\gamma, \delta > 0$, and define $Q = mu/(m+u)$. Then, with probability at least $1-\delta$ over all training and testing partitions, we have* $\mathcal{R}_u(f) \leq \mathcal{R}_m^\gamma(f) + \frac{2}{\gamma}\mathcal{O}\left(\epsilon\sqrt{Q\ln(\delta^{-1})}\right) + \mathcal{O}\left(\frac{\ln(\delta^{-1})}{\sqrt{Q}}\right)$.

Recall that as we discussed in Section 4, deeper GCN is provable more expressive and can achieve very low training error $\mathcal{R}_m^\gamma(f)$ if properly training. Then, if the dataset size is sufficiently large, the testing error $\mathcal{R}_u(f)$ will be dominated by the generalization gap, which is mainly controlled by uniformly stable constant ϵ. In the following, we explore the impact of GCN structures on ϵ. Our key idea is to decompose the ϵ into three terms: the Lipschitz continuous constant ρ_f, upper bound on gradient G_f, and the smoothness constant L_f of GCNs. Please refer to Lemma 1 for details.

Lemma 1. *Suppose function $f(\mathbf{h}^{(L)})$ is ρ_f-Lipschitz continuous, L_f-smooth, and the gradient of loss w.r.t. the parameter is bounded by G_f. After T steps of full-batch gradient descent, we have* $\epsilon = \frac{2\eta\rho_f G_f}{m}\sum_{t=1}^{T}(1+\eta L_f)^{t-1}$.

The proof is deferred to Appendix K. By using Lemma 1, we can derive constant ϵ of different model structures by comparing the Lipschitz continuity, smoothness, and gradient scale.

Before proceeding to our result, we make the following standard assumption on the node feature vectors and weight matrices, which are previously used in generalization analysis of GCNs [17, 35].

Assumption 1. *We assume the norm of node feature vectors, weight parameters are bounded, i.e., $\|\mathbf{x}_i\|_2 \leq B_x$, $\|\mathbf{W}^{(\ell)}\|_2 \leq B_w$, and $\|\mathbf{v}\|_2 \leq 1$.*

In Theorem 4, we show that the generalization bounds of GCN and its variants are dominated by the following terms: maximum node degree d, model depth L, training/validation set size (m, u), training iterations T, and spectral norm of the weight matrices B_w. The larger the aforementioned variables are, the larger the generalization gap is. We defer the formal statements and proofs to Appendices F, G, H, and I.

Theorem 4 (Informal)**.** *We say model is ϵ-uniformly stable with $\epsilon = \frac{2\eta\rho_f G_f}{m}\sum_{t=1}^{T}(1+\eta L_f)^{t-1}$ where the result of ρ_f, G_f, L_f are summarized in Table 1, and other related constants as*

$$\begin{aligned} B_d^\alpha &= (1-\alpha)\textstyle\sum_{\ell=1}^{L}(\alpha\sqrt{d})^{\ell-1} + (\alpha\sqrt{d})^L,\ B_w^\beta = \beta B_w + (1-\beta), \\ B_{\ell,d}^{\alpha,\beta} &= \max\left\{\beta\big((1-\alpha)L + \alpha\sqrt{d}\big), (1-\alpha)LB_w^\beta + 1\right\}. \end{aligned} \tag{1}$$

In the following, we provide intuitions and discussions on the generalization bound of each algorithm:

- Deep **GCN** requires iterations T to achieve small training error. Since the generalization bound increases with T, more iterations significantly hurt its generalization power. Note that our results considers both $B_w \leq 1$ and $B_w > 1$, where increasing model depth will not hurt the generalization if $B_w \leq 1$, and the generalization gap becomes sensitive to the model depth if $B_w > 1$. Notice that $B_w > 1$ is more likely to happen during training as we discussed in Section 4.
- **ResGCN** resolves the training difficulties by adding skip-connections between hidden layers. Although it requires less training iterations T, adding skip-connections enlarges the dependency

[6] Notice that a different loss function is used in convergence analysis (i.e., $\mathcal{L}(\boldsymbol{\theta})$ in Theorem 7) and generalization analysis ($\mathcal{R}_m^\gamma(f)$ in Theorem 4). We do this because current understanding on the convergence of deep neural networks is still mostly limited to the square loss, but the margin loss is a more suitable and widely accepted loss function for generalization analysis.

Table 1: Comparison of uniform stability constant ϵ of GCN variants, where $\mathcal{O}(\cdot)$ is used to hide constants that shared between all bounds.

	ρ_f and G_f	L_f	C_1 and C_2
ϵ_{GCN}	$\mathcal{O}(C_1^L C_2)$	$\mathcal{O}\big(C_1^L C_2\big((L+2)C_1^L C_2 + 2\big)\big)$	$C_1 = \max\{1, \sqrt{d}B_w\},\ C_2 = \sqrt{d}(1+B_x)$
ϵ_{ResGCN}	$\mathcal{O}(C_1^L C_2)$	$\mathcal{O}\big(C_1^L C_2\big((L+2)C_1^L C_2 + 2\big)\big)$	$C_1 = 1 + \sqrt{d}B_w,\ C_2 = \sqrt{d}(1+B_x)$
ϵ_{APPNP}	$\mathcal{O}(C_1)$	$\mathcal{O}\big(C_1(C_1C_2)+1\big)$	$C_1 = B_d^\alpha B_x, C_2 = \max\{1, B_w\}$
ϵ_{GCNII}	$\mathcal{O}(\beta C_1^L C_2)$	$\mathcal{O}\big(\alpha\beta C_1^L C_2\big((\alpha\beta L+2)C_1^L C_2 + 2\beta\big)\big)$	$C_1 = \max\{1, \alpha\sqrt{d}B_w^\beta\},\ C_2 = \sqrt{d} + B_{\ell,d}^{\alpha,\beta} B_x$
ϵ_{DGCN}	$\mathcal{O}(C_1)$	$\mathcal{O}(C_1(C_1C_2)+1)$	$C_1 = (\sqrt{d})^L B_x, C_2 = \max\{1, B_w\}$

on the number of layers L and the spectral norm of weight matrices B_w, therefore results in a larger generalization gap and a poor generalization performance..

- **APPNP** alleviates the aforementioned dependency by decoupling the weight parameters and feature propagation. As a result, its generalization gap does not significantly change as L and B_w increase. The optimal α that minimizes the generalization gap can be obtained by finding the α that minimize the term B_d^α. Although APPNP can significantly reduce the generalization gap, because a single weight matrix is shared between all layers, its expressive power is not enough for large-scale challenging graph datasets [23].
- To gain expressiveness, **GCNII** proposes to add the weight matrices back and add another hyper-parameter that explicitly controls the dependency on B_w. Although GCNII achieves the state-of-the-art performances on several graph datasets, the selection of hyper-parameters is non-trivial compared to APPNP because α, β are coupled with L, B_w, and d. In practice, [6] builds a very deep GCNII by choosing β dynamically decreases as the number of layers and different α values for different datasets.
- By property chosen hyper-parameters, we have the following order on the generalization gap given the same training iteration T: APPNP $\leq$ GCNII $\leq$ GCN $\leq$ ResGCN, which exactly match our empirical evaluation on the generalization gap in Section 7 and Appendix E.

Remark 1. *It is worthy to provide an alternative view of DropEdge [46] and PairNorm [60] algorithms from a generalization perspective. To improve the generalization power of standard GCNs, DropEdge randomly drops edges in the training phase, which leads to a smaller maximum node degree $d_s < d$. PairNorm applies normalization on intermediate node embeddings to ensure that the total pairwise feature distances remain constant across layers, which leads to less dependency on d and B_w. However, since deep GCN requires significantly more iterations to achieve low training error than shallow one, the performance of applying DropEdge and PairNorm on GCNs is still degrading as the number of layers increases. Most importantly, our empirical results in Appendix E.3 and E.4 suggest that applying Dropout and PairNorm is hurting the training accuracy (i.e., not alleviating over-smoothing) but reducing the generalization gap.*

6 Decoupled GCN

We propose to decouple the expressive power from generalization ability with **D**ecoupled **GCN** (DGCN). The DGCN model can be mathematically formulated as $\mathbf{Z} = \sum_{\ell=1}^{L} \alpha_\ell f^{(\ell)}(\mathbf{X})$ and $f^{(\ell)}(\mathbf{X}) = \mathbf{P}^\ell \mathbf{X}\big(\beta_\ell \mathbf{W}^{(\ell)} + (1-\beta_\ell)\mathbf{I}\big)$, where $\mathbf{W}^{(\ell)}$, α_ℓ and β_ℓ are the learnable weights for ℓth layer function $f^{(\ell)}(\mathbf{X})$. The design of DGCN has the following key ingredients:

- (Decoupling) The generalization gap in GCN grows exponentially with the number of layers. To overcome this issue, we propose to decouple the weight matrices from propagation by assigning weight $\mathbf{W}^{(\ell)}$ to each individual layerwise function $f^{(\ell)}(\mathbf{X})$. DGCN can be thought of as an ensemble of multiple SGCs [53] with depth from 1 to L. By doing so, the generalization gap has less dependency on the number of weight matrices, and deep models with large receptive fields can incorporate information of the global graph structure. Please refer to Theorem 5 for the details.
- (Learnable α_ℓ) After decoupling the weight matrices from feature propagation, the layerwise function $f^{(\ell)}(\mathbf{X})$ with more propagation steps can suffer from less expressive power. Therefore, we propose to assign a learnable weight α_ℓ for each step of feature propagation. Intuitively, DGCN assigns smaller weight α_ℓ to each layerwise function $f^{(\ell)}(\mathbf{X})$ with more propagation

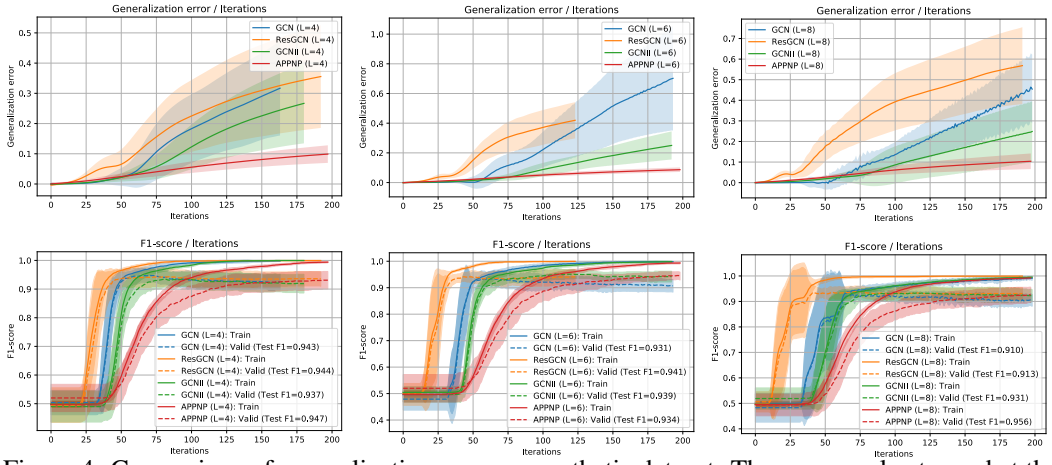

Figure 4: Comparison of generalization error on synthetic dataset. The curve early stopped at the largest training accuracy iteration.

steps at the beginning of training. Throughout the training, DGCN gradually adjusts the weight to leverage more useful large receptive field information.

- (Learnable β_ℓ) A learnable weight $\beta_\ell \in [0, 1]$ is assigned to each weight matrix to balance the expressiveness with model complexity, which guarantees a better generalization ability.

Theorem 5. *Let suppose α_ℓ and β_ℓ are pre-selected and fixed during training. We say DGCN is $\epsilon_{\texttt{DGCN}}$-uniformly stable with $\epsilon_{\texttt{DGCN}} = \frac{2\eta\rho_f G_f}{m}\sum_{t=1}^{T}(1+\eta L_f)^{t-1}$ where*

$$\rho_f = G_f = \mathcal{O}\Big((\sqrt{d})^L B_x\Big), L_f = \mathcal{O}\Big((\sqrt{d})^L B_x\big((\sqrt{d})^L B_x \max\{1, B_w\} + 1)\big)\Big). \tag{2}$$

The details are deferred to Appendix J, and comparison of bound to other GCN variants are summarized in Table 1. Depending on the automatic selection of α_ℓ, β_ℓ, the generalization bound of DGCN is between APPNP and GCN. In the following, we make connection to many GCN structures:

- **Connections to APPNP:** APPNP can be thought of as a variant of DGCN. More specifically, the layerwise weight in APPNP is computed as $\alpha_\ell = \alpha(1-\alpha)^\ell$ for $\ell < L$ and $\alpha_\ell = (1-\alpha)^\ell$ for $\ell = L$ given some constant $\alpha \in (0, 1)$, and the weight matrix is shared between all layers. Although DGCN has L weight matrices, its generalization is independent of the number of weight matrices, and thus enjoys a low generalization error with high expressiveness.
- **Connections to GCNII:** GCNII can be regarded as a variant of DGCN. Compared to GCNII, the decoupled propagation of DGCN significantly reduces the dependency of generalization error to the weight matrices. Besides, the learnable weights α_ℓ and β_ℓ allow DGCN to automatically adapt to challenging large-scale datasets without time-consuming hyper-parameter selection.
- **Connections to ResGCN:** By expanding the forward computation of ResGCN, we know that ResGCN can be think of as training an ensemble of GCNs from 1 to L layer, i.e., $\mathbf{H}^{(L)} = \sum_{\ell=1}^{L}\alpha_\ell\sigma(\mathbf{P}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)})$ with $\alpha_\ell = 1$. In other word, ResNet can be regarded as the "*summation* of the model complexity" of L-layer. However, DGCN is using $\sum_{\ell=1}^{L}\alpha_\ell = 1$, which can be thought of as a "*weighted average* of model complexity". Therefore, ResGCN is a special case of DGCN with equal weights α_ℓ on each layerwise function. With just a simple change on the ResNet structure, our model DGCN is both easy to train and good to generalize.

7 Experiments

Synthetic dataset. We empirically compare the generalization error of different GCN structures on the synthetic dataset. In particular, we create the synthetic dataset by contextual stochastic block model (CSBM) [12] with two equal-size classes. CSBM is a graph generation algorithm that adds Gaussian random vectors as node features on top of classical SBM. CSBM allows for smooth control over the information ratio between node features and graph topology by a pair of hyper-parameter (μ, λ), where μ controls the diversity of the Gaussian distribution and λ controls the number of edges between intra- and inter-class nodes. We generate random graphs with 1000 nodes, average node

degree as 5, and each node has a Gaussian random vector of dimension 1000 as node features. We chose 75% nodes as training set, 15% of nodes as validation set for hyper-parameter tuning, and the remaining nodes as testing set. We conduct an experiment 20 times by randomly selecting (μ, λ) such that both node feature and graph topology are equally informative.

As shown in Figure 4, we have the following order on the generalization gap given the same training iteration T: APPNP $\leq$ GCNII $\leq$ GCN $\leq$ ResGCN, which exactly match the theoretical result in Theorem 4. More specifically, ResGCN has the largest generalization gap due to the skip-connections, APPNP has the smallest generalization gap by removing the weight matrices in each individual layer. GCNII achieves a good balance between GCN and APPNP by balancing the expressive and generalization power. Finally, DGCN enjoys a small generalization error by using the decoupled GCN structure.

Open graph benchmark dataset. As pointed out by Hu et al. [23], the traditional commonly-used graph datasets are unable to provide a reliable evaluation due to various factors including dataset size, leakage of node features and no consensus on data splitting. To truly evaluate the expressive and the generalization power of existing methods, we evaluate on the open graph benchmark (OGB) dataset. Experiment setups are based on the default setting for GCN implementation on the leaderboard. We choose the hidden dimension as 128, learning rate as 0.01, dropout ratio as 0.5 for *Arxiv* dataset, and no dropout for *Products* and *Protein* datasets. We train $300/1000/500$ epochs for *Products*, *Proteins*, and *Arxiv* dataset respectively. Due to limited GPU memory, the number of layers is selected as the one with the best performance between 2 to 16 layers for *Arxiv* dataset, 2 to 8 layers for *Protein* dataset, and 2 to 4 for *Products* dataset. We choose α_ℓ from $\{0.9, 0.8, 0.5\}$ for APPNP and GCNII, and use $\beta_\ell = 0.5/\ell$ for GCNII, and select the setup with the best validation result for comparison.

As shown in Table 2, DGCN achieves a compatible performance to GCNII[7] without the need of manually tuning the hyper-parameters for all settings, and it significantly outperform APPNP and ResGCN. Due to the space limit, the detailed setups and more results can be found in Appendix E. Notice that generalization bounds are more valuable when comparing two models with same training accuracy (therefore we first show in Section 4 that deeper model can also achieve low training error before our discussion on generalization in Section 5). In Table 2, because ResGCN has no restriction on the weight matrices, it can achieve lower training error and its test performance is mainly restricted by its generalization error. However, because GCNII and APPNP have restrictions on the weight matrices, their performance is mainly restricted by their training error. A model with small generalization error and no restriction on the weight (e.g., DGCN) is preferred as it has higher potential to reach a better test accuracy by reducing its training error.

Table 2: Comparison of F1-score on OGB dataset.

%	**Products**	**Proteins**	**Arvix**
GCN	75.39 ± 0.21	71.66 ± 0.48	71.56 ± 0.19
ResGCN	75.53 ± 0.12	74.50 ± 0.41	72.56 ± 0.31
APPNP	66.35 ± 0.10	71.78 ± 0.29	68.02 ± 0.55
GCNII	$71.93 \pm 0.35^\dagger$	75.60 ± 0.47	$72.57 \pm 0.23^\ddagger$
DGCN	76.09 ± 0.29	75.45 ± 0.24	72.63 ± 0.12

8 Conclusion

In this work, we show that there exists a discrepancy between over-smoothing based theoretical results and the practical behavior of deep GCNs. Our theoretical result shows that a deeper GCN can be as expressive as a shallow GCN, if it is properly trained. To truly understand the performance decay issue of deep GCNs, we provide the first transductive uniform stability-based generalization analysis of GCNs and other GCN structures. To improve the optimization issue and benefit from depth, we propose DGCN that enjoys a provable high expressive power and generalization power. We conduct empirical evaluations on various synthetic and real-world datasets to validate the correctness of our theory and advantages over the baselines.

Acknowledgements

This work was supported in part by NSF grant 2008398.

[7] † GCNII underfits with the default hyper-parameters. ‡ GCNII achieves 72.74 ± 0.16 by using hidden dimension as 256 and a different design of graph convolution layer. Refer here for details.

References

[1] Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. In *International Conference on Learning Representations*, 2020.

[2] Rianne van den Berg, Thomas N Kipf, and Max Welling. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*, 2017.

[3] Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of machine learning research*, 2(Mar):499–526, 2002.

[4] Chen Cai and Yusu Wang. A note on over-smoothing for graph neural networks. *arXiv preprint arXiv:2006.13318*, 2020.

[5] Lei Chen, Zhengdao Chen, and Joan Bruna. On graph neural networks versus graph-augmented mlps. In *International Conference on Learning Representations*, 2020.

[6] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1725–1735. PMLR, 2020.

[7] Zhengdao Chen, Lei Chen, Soledad Villar, and Joan Bruna. Can graph neural networks count substructures? In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[8] Weilin Cong, Rana Forsati, Mahmut T. Kandemir, and Mehrdad Mahdavi. Minimal variance sampling with provable guarantees for fast training of graph neural networks. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1393–1403. ACM, 2020.

[9] Zhiyong Cui, Kristian Henrickson, Ruimin Ke, and Yinhai Wang. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[10] Kenneth R Davidson and Stanislaw J Szarek. Local operator theory, random matrices and banach spaces. *Handbook of the geometry of Banach spaces*, 1(317-366):131, 2001.

[11] Songgaojun Deng, Huzefa Rangwala, and Yue Ning. Learning dynamic context graphs for predicting social events. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 1007–1016. ACM, 2019.

[12] Yash Deshpande, Subhabrata Sen, Andrea Montanari, and Elchanan Mossel. Contextual stochastic block models. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8590–8602, 2018.

[13] Kien Do, Truyen Tran, and Svetha Venkatesh. Graph transformation policy network for chemical reaction prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 750–760. ACM, 2019.

[14] Simon S. Du, Kangcheng Hou, Ruslan Salakhutdinov, Barnabás Póczos, Ruosong Wang, and Keyulu Xu. Graph neural tangent kernel: Fusing graph neural networks with graph kernels. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5724–5734, 2019.

[15] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2224–2232, 2015.

[16] Ran El-Yaniv and Dmitry Pechyony. Stable transductive learning. In *Learning Theory, 19th Annual Conference on Learning Theory, COLT 2006, Pittsburgh, PA, USA, June 22-25, 2006, Proceedings*, volume 4005 of *Lecture Notes in Computer Science*, pages 35–49. Springer, 2006.

[17] Vikas K. Garg, Stefanie Jegelka, and Tommi S. Jaakkola. Generalization and representational limits of graph neural networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3419–3430. PMLR, 2020.

[18] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[19] Shunwang Gong, Mehdi Bahri, Michael M. Bronstein, and Stefanos Zafeiriou. Geometrically principled connections in graph neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 11412–11421. IEEE, 2020.

[20] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1024–1034, 2017.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1026–1034. IEEE Computer Society, 2015.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[23] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[24] Wei Huang, Yayong Li, Weitao Du, Richard Yi Da Xu, Jie Yin, and Ling Chen. Wide graph neural networks: Aggregation provably leads to exponentially trainability loss. *arXiv preprint arXiv:2103.03113*, 2021.

[25] Wenbing Huang, Yu Rong, Tingyang Xu, Fuchun Sun, and Junzhou Huang. Tackling over-smoothing for general graph convolutional networks. *arXiv e-prints*, pages arXiv–2008, 2020.

[26] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[27] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[28] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[29] Kezhi Kong, Guohao Li, Mucong Ding, Zuxuan Wu, Chen Zhu, Bernard Ghanem, Gavin Taylor, and Tom Goldstein. Flag: Adversarial data augmentation for graph neural networks. *arXiv preprint arXiv:2010.09891*, 2020.

[30] Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 1269–1278. ACM, 2019.

[31] AA Leman and B Weisfeiler. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsiya*, 2(9):12–16, 1968.

[32] Guohao Li, Matthias Müller, Ali K. Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 9266–9275. IEEE, 2019.

[33] Jia Li, Zhichao Han, Hong Cheng, Jiao Su, Pengyun Wang, Jianfeng Zhang, and Lujia Pan. Predicting path failure in time-evolving graphs. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 1279–1289. ACM, 2019.

[34] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3538–3545. AAAI Press, 2018.

[35] Renjie Liao, Raquel Urtasun, and Richard Zemel. A pac-bayesian approach to generalization bounds for graph neural networks. *arXiv preprint arXiv:2012.07690*, 2020.

[36] Andreas Loukas. What graph neural networks cannot learn: depth vs width. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[37] Sitao Luan, Mingde Zhao, Xiao-Wen Chang, and Doina Precup. Training matters: Unlocking potentials of deeper graph convolutional neural networks. *arXiv preprint arXiv:2008.08838*, 2020.

[38] Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 4602–4609. AAAI Press, 2019.

[39] Quynh Nguyen. On the proof of global convergence of gradient descent for deep relu networks with linear widths. *arXiv preprint arXiv:2101.09612*, 2021.

[40] Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[41] 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, 2020.

[42] Namyong Park, Andrey Kan, Xin Luna Dong, Tong Zhao, and Christos Faloutsos. Estimating node importance in knowledge graphs using graph neural networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 596–606. ACM, 2019.

[43] Jiezhong Qiu, Jian Tang, Hao Ma, Yuxiao Dong, Kuansan Wang, and Jie Tang. Deepinf: Modeling influence locality in large social networks. In *KDD*, 2018.

[44] Afshin Rahimi, Trevor Cohn, and Timothy Baldwin. Semi-supervised user geolocation via graph convolutional networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2009–2019. Association for Computational Linguistics, 2018.

[45] Morteza Ramezani, Weilin Cong, Mehrdad Mahdavi, Anand Sivasubramaniam, and Mahmut Kandemir. Gcn meets gpu: Decoupling "when to sample" from "how to sample". *Advances in Neural Information Processing Systems*, 33, 2020.

[46] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[47] Ruslan Salakhutdinov. Deep learning. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, page 1973. ACM, 2014.

[48] Franco Scarselli, Ah Chung Tsoi, and Markus Hagenbuchner. The vapnik–chervonenkis dimension of graph and recursive neural networks. *Neural Networks*, 108:248–259, 2018.

[49] 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, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 1539–1548. ACM, 2019.

[50] Hao Wang, Tong Xu, Qi Liu, Defu Lian, Enhong Chen, Dongfang Du, Han Wu, and Wen Su. MCNE: an end-to-end framework for learning multiple conditional network representations of social network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 1064–1072. ACM, 2019.

[51] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 968–977. ACM, 2019.

[52] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. KGAT: knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 950–958. ACM, 2019.

[53] Felix Wu, Amauri H. Souza Jr., Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. Simplifying graph convolutional networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6861–6871. PMLR, 2019.

[54] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[55] Keyulu Xu, Jingling Li, Mozhi Zhang, Simon S. Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. What can neural networks reason about? In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[56] Keyulu Xu, Mozhi Zhang, Jingling Li, Simon Shaolei Du, Ken-Ichi Kawarabayashi, and Stefanie Jegelka. How neural networks extrapolate: From feedforward to graph neural networks. In *International Conference on Learning Representations*, 2021.

[57] Yujun Yan, Milad Hashemi, Kevin Swersky, Yaoqing Yang, and Danai Koutra. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. *arXiv preprint arXiv:2102.06462*, 2021.

[58] Chaoqi Yang, Ruijie Wang, Shuochao Yao, Shengzhong Liu, and Tarek Abdelzaher. Revisiting" over-smoothing" in deep gcns. *arXiv preprint arXiv:2003.13663*, 2020.

[59] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 974–983. ACM, 2018.

[60] Lingxiao Zhao and Leman Akoglu. Pairnorm: Tackling oversmoothing in gnns. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[61] Kuangqi Zhou, Yanfei Dong, Wee Sun Lee, Bryan Hooi, Huan Xu, and Jiashi Feng. Effective training strategies for deep graph neural networks. *arXiv preprint arXiv:2006.07107*, 2020.

[62] Xianchen Zhou and Hongxia Wang. The generalization error of graph convolutional networks may enlarge with more layers. *Neurocomputing*, 424:97–106, 2021.

Supplementary Material
On Provable Benefits of Depth in Training Graph Convolutional Networks

Organization. In Section A, we provide additional empirical evaluations on the change of pairwise distances for node embeddings as the number of layers increases. In Section B, we summarize the existing theoretical results on over-smoothing and provide empirical validation on whether over-smoothing happens in practice. In Sections C and D, we provide the proof of Theorem 1 (expressive power) and Theorem 2 (convergence to global optimal and characterization of number of iterations), respectively. In Section E, we provide more empirical results on the effectiveness of the proposed algorithm. In Sections F, G, H, I, and J we provide the generalization analysis of GCN, ResGCN, APPNP, GCNII, and DGCN, respectively. Code can be found at the following repository:

https://github.com/CongWeilin/DGCN.

A Empirical results on the pairwise distance of node embeddings

In this section, we provide additional empirical evaluations on the change of pairwise distance for node embeddings. First we introduce a concrete definition of the intra- and inter-class pairwise distances, then provide the experimental setups, and finally illustrate the results.

Definition of pairwise distance. To verify whether over-smoothing exists in GCNs, we define the intra-class pairwise distance of $\mathbf{H}^{(\ell)}$ as the average pairwise Euclidean distance of two-node embeddings if they have the same ground truth label. Similarly, we define the inter-class pairwise distance of $\mathbf{H}^{(\ell)}$ as the average pairwise distance of two node embeddings if they have different ground truth labels.

$$\text{intra}(\mathbf{H}^{(\ell)}) \triangleq \frac{\sum_{i,j=1}^{N} \mathbf{1}\{y_i = y_j\} \cdot \|\boldsymbol{h}_i^{(\ell)} - \boldsymbol{h}_j^{(\ell)}\|_2^2}{\|\mathbf{H}^{(\ell)}\|_F \cdot \sum_{i,j=1}^{N} \mathbf{1}\{y_i = y_j\}},$$
$$\text{inter}(\mathbf{H}^{(\ell)}) \triangleq \frac{\sum_{i,j=1}^{N} \mathbf{1}\{y_i \neq y_j\} \cdot \|\boldsymbol{h}_i^{(\ell)} - \boldsymbol{h}_j^{(\ell)}\|_2^2}{\|\mathbf{H}^{(\ell)}\|_F \cdot \sum_{i,j=1}^{N} \mathbf{1}\{y_i \neq y_j\}}.$$

Both intra- and inter-class distances are normalized using the Frobenius norm to eliminate the difference caused by the scale of the node embedding matrix. By the definition of over-smoothing in [34], the distance between intra- and inter-class pairwise distance should be decreasing sharply as the number of GCN layers increases.

Setup. In Figure 5 and Figure 6, we plot the training error in a 10-layer GCN [27], SGC [53], and APPNP [28] models until convergence, and choose the model with the best validation score for pairwise distance computation. The pairwise distance at the ℓth layer is computed by the node embedding matrix $\mathbf{H}^{(\ell)}$ generated by the selected model. We repeat each experiment 10 times and plot the mean and standard deviation of the results.

Results. As shown in Figure 5 and Figure 6, when ignoring the weight matrices, i.e., sub-figures in box (b), the intra- and inter-class pairwise distance are decreasing but the gap between intra- and inter-class pairwise distance is increasing as the model goes deeper. That is, the difference between intra- and inter-class nodes' embeddings is increasing and becoming more discriminative. On the other hand, when considering the weight matrices, i.e., sub-figures in box (a), the difference between intra- and inter-class node embeddings is large and increasing as the model goes deeper. In other words, weight matrices learn to make node embeddings discriminative and generate expressive node embeddings during training.

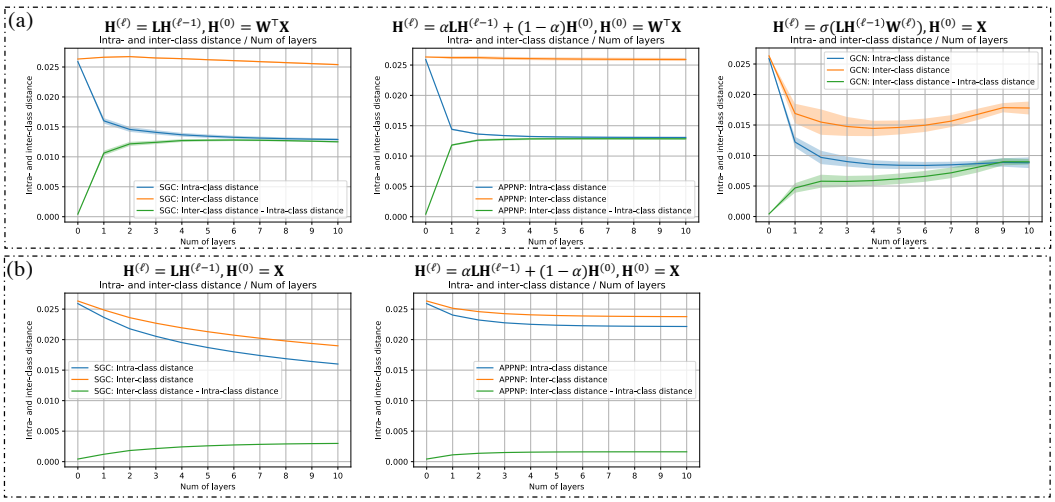

Figure 5: Comparison of the pairwise distance for intra- and inter-class node embeddings on *Cora* dataset for different models by increasing the number of layers in the proposed architecture.

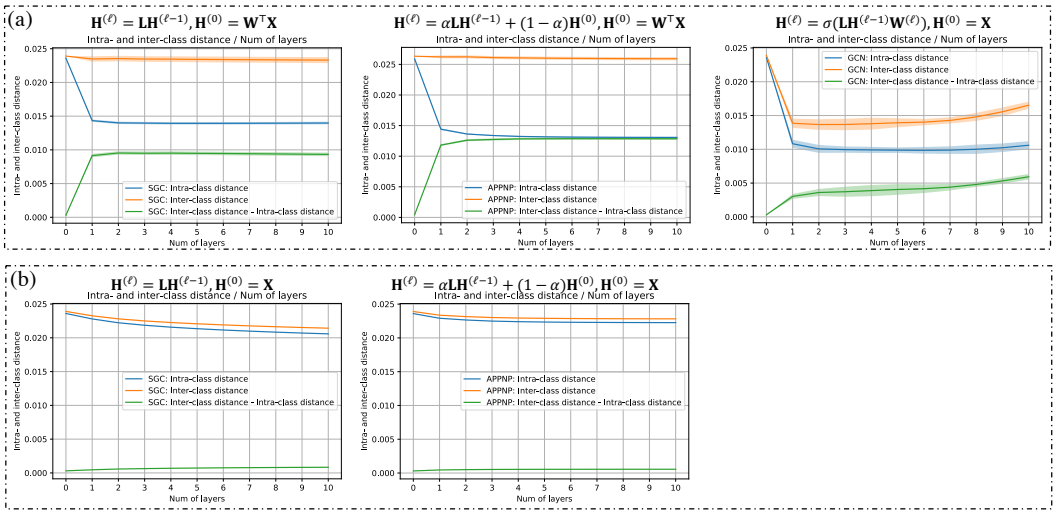

Figure 6: Comparison of the pairwise distance for intra- and inter-class node embeddings on *Citeseer* dataset for different models by increasing the number of layers in the proposed architecture.

B A summary of theoretical results on over-smoothing

In this section, we survey the existing theories on understanding over-smoothing in GCNs from [25, 40, 4] and illustrate why the underlying assumptions may not hold in practice. Finally, we discuss conditions where over-smoothing and over-fitting might happen simultaneously.

Before processing, let first recall the notation we defined in Section 3. Recall that $\mathbf{A} \in \mathbb{R}^{N\times N}$ denotes the adjacency matrix of the self-connected graph, i.e., $A_{i,j} = 1$ if edge $(i,j) \in \mathcal{E}$ and $A_{i,j} = 0$ otherwise, $\mathbf{D} \in \mathbb{R}^{N\times N}$ denotes the corresponding degree matrix, i.e., $D_{i,i} = \deg(i)$ and $D_{i,j} = 0$ if $i \neq j$, and the symmetric Laplacian matrix is computed as $\mathbf{L} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$.

B.1 Expressive power based analysis [41, 25]

Proposition 1 (Proposition 1 in [40], Theorem 1 of [25])**.** *Let $\lambda_1 \geq \ldots \geq \lambda_N$ denote the eigenvalues of the Laplacian matrix $\mathbf{L}$ in descending order, and let $\mathbf{e}_i$ denote the eigenvector associated with the eigenvalue λ_i. Suppose graph has $M \leq N$ connected components, then we have $\lambda_1 = \ldots = \lambda_M = 1$*

and $1 > \lambda_{M+1} > \ldots > \lambda_N$. *Let* $\mathbf{E} = \{\mathbf{e}_i\}_{i=1}^M \in \mathbb{R}^{M\times N}$ *denote the stack of eigenvectors that associated to eigenvalues* $\lambda_1, \ldots, \lambda_M$. *Then for all* $i \in [M]$, *the eigenvector* $\mathbf{e}_i \in \mathbb{R}^N$ *is defined as*

$$\mathbf{e}_i = \{e_i(j)\}_{j=1}^N,\ e_i(j) \propto \begin{cases} \sqrt{\deg(j)} & \textit{if } j\textit{th node is in the } i\textit{th connected component} \\ 0 & \textit{otherwise} \end{cases}. \tag{3}$$

Proof. The proof can be found in Section B of [40]. □

[40] proposes to measure the expressive power of node embeddings using its distance to a subspace $\mathcal{M}$ that only has node degree information, i.e., regardless of the node feature information.

Definition 2 (Subspace and distance to subspace)**.** *We define subspace* $\mathcal{M}$ *as*

$$\mathcal{M} := \{\mathbf{E}^\top \mathbf{R} \in \mathbb{R}^{N\times d} \mid \mathbf{R} \in \mathbb{R}^{M\times d}\}, \tag{4}$$

where $\mathbf{E} = \{\mathbf{e}_i\}_{i=1}^M \in \mathbb{R}^{M\times N}$ *is the orthogonal basis of Laplacian matrix* $\mathbf{L}$, *and* $\mathbf{R}$ *is any random matrix. The distance of a node embedding matrix* $\mathbf{H}^{(\ell)} \in \mathbb{R}^{N\times d}$ *to the subspace can be computed as* $d_{\mathcal{M}}(\mathbf{H}^{(\ell)}) = \inf_{\mathbf{Y}\in\mathcal{M}} \|\mathbf{H}^{(\ell)} - \mathbf{Y}\|_{\mathrm{F}}$, *where* $\|\cdot\|_{\mathrm{F}}$ *denotes the Frobenius norm of the matrix.*

Subspace $\mathcal{M}$ is a d-dimensional subspace defined by orthogonal vectors $\{\mathbf{e}_i\}_{i=1}^M$ that only captures node degree information. Intuitively, a smaller distance $d_{\mathcal{M}}(\mathbf{H}^{(\ell)})$ means that $\mathbf{H}^{(\ell)}$ is losing its feature information but only carries the degree information.

In the following, we summarize the theoretical results of over-smoothing in [25] and extend their results to SGC and GCNII, which will be used in our discussion below.

Theorem 6 (Theorem 2 of [25])**.** *Let* $\lambda = \max_{i\in[M+1,N]} |\lambda_i|$ *denote the largest absolute eigenvalue of Laplacian matrix* $\mathbf{L}$ *which is bounded by* 1, *let* s *denote the largest singular-value of weight parameter* $\mathbf{W}^{(\ell)},\ \ell \in [L]$, *then we have*

$$d_{\mathcal{M}}(\mathbf{H}^{(\ell)}) - \varepsilon \le \gamma\Big(d_{\mathcal{M}}(\mathbf{H}^{(\ell-1)}) - \varepsilon\Big), \tag{5}$$

where γ *and* ε *are functions of* λ *and* s *that depend on the model structure.*

GCN. For vanilla *GCN* model with the graph convolutional operation defined as

$$\mathbf{H}^{(\ell)} = \sigma(\mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)}),\ \mathbf{H}^{(0)} = \mathbf{X}, \tag{6}$$

we have $\gamma_{\mathsf{GCN}} = \lambda s$ and $\epsilon_{\mathsf{GCN}} = 0$.

Under the assumption that $\gamma_{\mathsf{GCN}} \le 1$, i.e., the graph is densely connected and the largest singular value of weight matrices is small, we have $d_{\mathcal{M}}(\mathbf{H}^{(\ell+1)}) \le \gamma_{\mathsf{GCN}} \cdot d_{\mathcal{M}}(\mathbf{H}^{(\ell)})$. In other words, the vanilla *GCN* model is losing expressive power as the number of layers increases.

However, if we suppose there exist a trainable bias parameter $\mathbf{B}^{(\ell)} = \{\mathbf{b}^{(\ell)}\}_{i=1}^N \in \mathbb{R}^{N\times d_\ell}$ in each graph convolutional layer

$$\mathbf{H}^{(\ell)} = \sigma(\mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)} + \mathbf{B}^{(\ell)}),\ \mathbf{H}^{(0)} = \mathbf{X}, \tag{7}$$

we have $\gamma_{\mathsf{GCN}} = \lambda s$ and $\epsilon_{\mathsf{GCN}} = d_{\mathcal{M}}(\mathbf{B}^{(\ell)})$. When $d_{\mathcal{M}}(\mathbf{B}^{(\ell)})$ is considerably large, there is no guarantee that *GCN-bias* suffers from losing expressive power issue.

ResGCN. For GCN with residual connection, the graph convolution operation is defined as

$$\mathbf{H}^{(\ell)} = \sigma(\mathbf{L}\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)}) + \mathbf{H}^{(\ell-1)},\ \mathbf{H}^{(0)} = \mathbf{X}\mathbf{W}^{(0)}, \tag{8}$$

where we have $\gamma_{\mathsf{ResGCN}} = 1 + \lambda s$ and $\epsilon_{\mathsf{ResGCN}} = 0$. Since $\gamma_{\mathsf{ResGCN}} \ge 1$, there is no guarantee that *ResGCN* suffers from losing expressive power.

APPNP. For *APPNP* with graph convolution operation defined as

$$\mathbf{H}^{(\ell)} = \alpha\mathbf{L}\mathbf{H}^{(\ell)} + (1-\alpha)\mathbf{H}^{(0)},\ \mathbf{H}^{(0)} = \mathbf{X}\mathbf{W}^{(0)}, \tag{9}$$

we have $\gamma_{\mathsf{APPNP}} = \alpha\lambda$ and $\epsilon_{\mathsf{APPNP}} = \frac{(1-\alpha)d_{\mathcal{M}}(\mathbf{H}^{(0)})}{1-\gamma_{\mathsf{APPNP}}}$. Although $\gamma_{\mathsf{APPNP}} < 1$, because $\epsilon_{\mathsf{APPNP}}$ can be large, there is no guarantee that *APPNP* suffers from losing expressive power.

GCNII. For *GCNII* with graph convolution operation defined as

$$\mathbf{H}^{(\ell)} = \sigma\Big(\big(\alpha\mathbf{L}\mathbf{H}^{(\ell)} + (1-\alpha)\mathbf{H}^{(0)}\big)\big(\beta\mathbf{W}^{(\ell)} + (1-\beta)\mathbf{I}_N\big)\Big),\ \mathbf{H}^{(0)} = \mathbf{X}\mathbf{W}^{(0)}, \tag{10}$$

we have $\gamma_{\mathsf{GCNII}} = \Big(1-(1-\beta)(1-s)\Big)\alpha\lambda$, and $\epsilon_{\mathsf{GCNII}} = \frac{(1-\alpha)d_{\mathcal{M}}(\mathbf{H}^{(0)})}{1-\gamma_{\mathsf{GCNII}}}$. Although we have $\gamma_{\mathsf{GCNII}} < 1$, because $\epsilon_{\mathsf{GCNII}}$ can be considerably large, there is no guarantee that *GCNII* suffers from losing expressive power.

SGC. The result can be also extended to the linear model *SGC* [53], where the graph convolution is defined as

$$\mathbf{H}^{(\ell)} = \mathbf{L}\mathbf{H}^{(\ell-1)},\ \mathbf{H}^{(0)} = \mathbf{X}\mathbf{W}^{(0)}, \tag{11}$$

we have $\gamma_{\mathsf{SGC}} = \lambda$, and $\epsilon_{\mathsf{SGC}} = 0$, which guarantees losing expressive power as the number of layers increases.

Discussion on the result of Theorem 6. In summary, the linear model *SGC* always suffers from losing expressive power without the assumption on the trainable parameters. Under the assumption that the multiplication of the *largest singular-value of trainable parameter* and *the largest absolute eigenvalue of Laplacian matrix smaller than* 1, i.e., $\lambda s \leq 1$, we can only guarantee that *GCN* suffers from losing expressive power issue, but cannot have the same guarantee on *GCN-bias*, *ResGCN*, *APPNP*, and *GCNII*. However, as we show in Figure 3, the assumption is not going to hold in practice, and the distances are not decreasing for most cases.

Remark 2. *[40] conducts experiments on Erdős-Rényi graph to show that when the graph is sufficiently dense and large, vanilla GCN suffers from expressive lower loss. Erdős-Rényi graph is constructed by connecting nodes randomly. Each edge is included in the graph with probability p independent from every other edge. To guarantee a small λ, a dense graph with larger p is required. For example, in the Section 6.2 of [40], they choose $p = 0.5$ (one node is connected to 50% of other nodes) such that $\lambda = 0.063$ and choose $p = 0.1$ (one node is connected to 10% of other nodes) such that $\lambda = 0.195$. However, real world datasets are sparse and have a λ that is closer to 1. For example, Cora has $\lambda = 0.9964$, Citeseer has $\lambda = 0.9987$, and Pubmed has $\lambda = 0.9905$.*

In Figure 7 and Figure 8, we compare the testing set F1-score, expressive power metric $d_{\mathcal{M}}(\mathbf{H}^{(\ell)})$ before and after training. $d_{\mathcal{M}}(\mathbf{H}^{(\ell)})$ is computed on *the final output of a ℓ-layer GCN model*. The "after training results" are computed on the model with the best testing score. We repeat each experiment 10 times and report the average values. As shown in Figure 7 and Figure 8, the expressive power metric $d_{\mathcal{M}}(\mathbf{H}^{(\ell)})$ of the *untrained* vanilla GCN is decreasing as the number of layers increases. However, the expressive power metric $d_{\mathcal{M}}(\mathbf{H}^{(\ell)})$ of the *trained* vanilla GCN increases with more layers. Besides, we observe that there is no obvious connection between the testing F1-score to $d_{\mathcal{M}}(\mathbf{H}^{(\ell)})$ for other GCN variant. For example, the testing F1-score of APPNP is increasing while its $d_{\mathcal{M}}(\mathbf{H}^{(\ell)})$ is not changing much.

In Figure 9 and Figure 10, we compare the testing set F1-score, expressive power metric $d_{\mathcal{M}}(\mathbf{H}^{(\ell)})$ before and after training. $d_{\mathcal{M}}(\mathbf{H}^{(\ell)})$ is computed on *the ℓth layer intermediate node embeddings of a* 10*-layer GCN model*. The "after training results" are computed on the model with the best testing score. We repeat experiment 10 times and report the average values. As shown in Figure 9 and Figure 10, the expressive power metric $d_{\mathcal{M}}(\mathbf{H}^{(\ell)})$ of the *untrained* vanilla GCN is decreasing as the number of layers increases. However, the expressive power metric $d_{\mathcal{M}}(\mathbf{H}^{(\ell)})$ of the *trained* vanilla GCN increases as the number of layers increases.

B.2 Dirichlet energy based analysis [4]

Definition 3 (Dirichlet energy)**.** *The Dirichlet energy of node embedding matrix $\mathbf{H}^{(\ell)}$ is defined as*

$$\mathit{DE}(\mathbf{H}^{(\ell)}) = \frac{1}{2}\sum_{i,j=1}^{N} L_{i,j}\left\|\frac{\mathbf{h}_i^{(\ell)}}{\sqrt{\deg(i)}} - \frac{\mathbf{h}_j^{(\ell)}}{\sqrt{\deg(j)}}\right\|_2^2. \tag{12}$$

Intuitively, the Dirichlet energy measures the smoothness of node embeddings. Based on the definition of Dirichlet energy, they obtain a similar result as shown in [41] that

$$\mathtt{DE}(\mathbf{H}^{(\ell+1)}) \leq \lambda s \cdot \mathtt{DE}(\mathbf{H}^{(\ell)}), \tag{13}$$

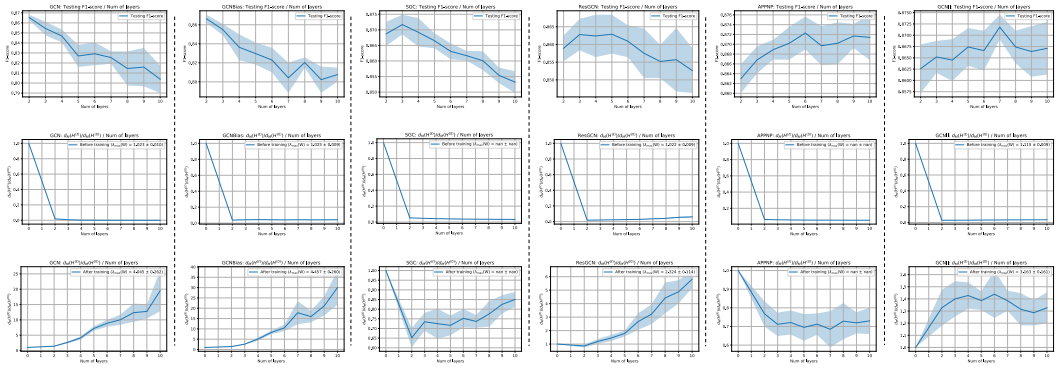

Figure 7: Comparison of F1-score, expressive power metric $d_{\mathcal{M}}(\mathbf{H}^{(\ell)})$ of *GCN*, *GCN-bias*, *SGC*, *ResGCN*, *APPNP*, and *GCNII* before and after training on *Cora* dataset. The average "largest sigular value" of weight matrices in graph convolutional layers are reported, where "nan" stands for no weight matrices inside graph convolution layers. We only consider GCN model with depth from 2-layers to 10-layers. (Better viewed in PDF version)

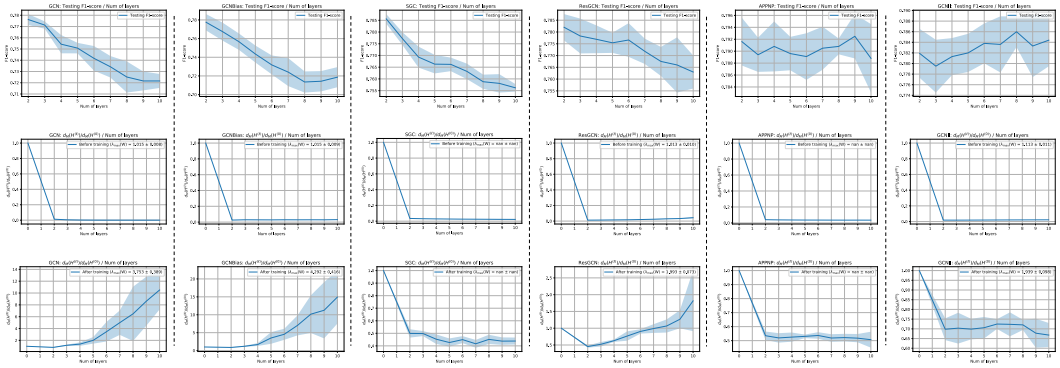

Figure 8: Comparison of F1-score, expressive power metric $d_{\mathcal{M}}(\mathbf{H}^{(\ell)})$ of *GCN*, *GCN-bias*, *SGC*, *ResGCN*, *APPNP*, and *GCNII* before and after training on *Citeseer* dataset. The average "largest singular value" of weight matrices in graph convolutional layers are reported, where "nan" stands for no weight matrices inside graph convolution layers. We only consider GCN model with depth from 2-layers to 10-layers. (Better viewed in PDF version)

where λ is *largest absolute eigenvalue of Laplacian matrix* $\mathbf{L}$ *that is less than* 1, and s is *the largest singular-value of trainable parameter* $\mathbf{W}^{(\ell)}$.

Remark 3. *The Dirichlet energy used in [4] is closely related to the pairwise distance as shown in Figure 2. In Figure 2, we measure the "smoothness" or "expressive power" of node embeddings using a normalized Dirichlet energy function, i.e.,* $\frac{\mathtt{DE}(\mathbf{H}^{(\ell)})}{\|\mathbf{H}^{(\ell)}\|_F^2}$ *for inner- and cross-class nodes respectively, which refers to the real smoothness metric of graph signal as pointed out in the footnote 1 of [4].*

Intuitively, under the assumption that the largest singular-value of weight matrices less than 1, as the number of layers increase, i.e., $\ell \to \infty$, the Dirichlet energy of node embeddings $\mathtt{DE}(\mathbf{H}^{(\ell)})$ is decreasing. However, recall that this assumption is not likely to hold in practice because the real-world graphs are usually sparse and the largest singular value of weight matrices is usually larger than 1.

Besides, generalizing the result to other GCN variants (e.g., ResGCN, GCNII) is non-trivial. For example in ResGCN, we have

$$\begin{aligned}\mathtt{DE}(\mathbf{H}^{(\ell)}) &= \mathtt{DE}(\sigma(\mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)}) + \mathbf{H}^{(\ell-1)}) \\ &\leq 2\mathtt{DE}(\sigma(\mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)})) + 2\mathtt{DE}(\mathbf{H}^{(\ell-1)}) \\ &\leq 2(\lambda s + 1)\mathtt{DE}(\mathbf{H}^{(\ell-1)}).\end{aligned} \tag{14}$$

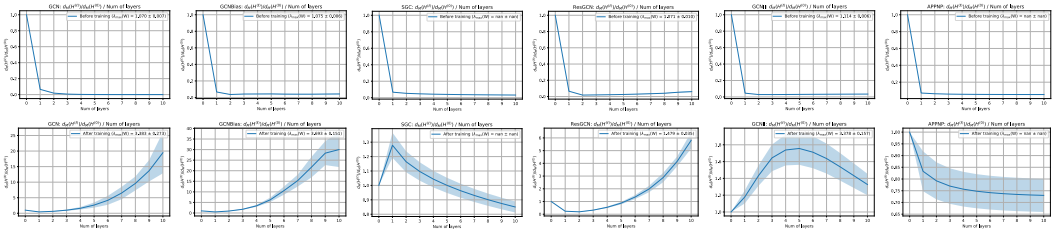

Figure 9: Comparison of F1-score, expressive power metric $d_{\mathcal{M}}(\mathbf{H}^{(\ell)})$ of 10-layer *GCN*, *GCN-bias*, *SGC*, *ResGCN*, *APPNP*, and *GCNII* before and after training on *Cora* dataset. The average "largest singular value" of weight matrices in graph convolutional layers are reported, where "nan" stands for no weight matrices inside graph convolution layers. (Better viewed in PDF version)

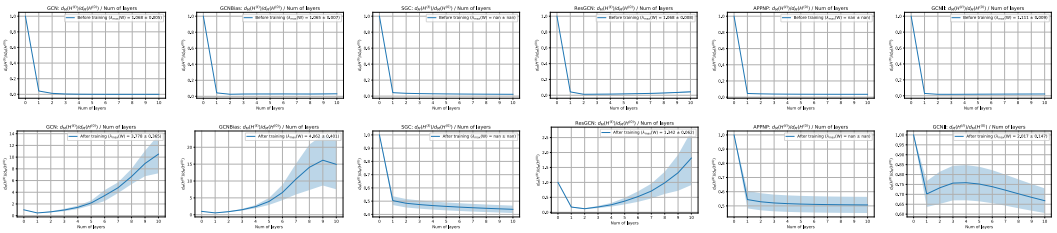

Figure 10: Comparison of F1-score, expressive power metric $d_{\mathcal{M}}(\mathbf{H}^{(\ell)})$ of 10-layer *GCN*, *GCN-bias*, *SGC*, *ResGCN*, *APPNP*, and *GCNII* before and after training on *Citeseer* dataset. The average "largest singular value" of weight matrices in graph convolutional layers are reported, where "nan" stands for no weight matrices inside graph convolution layers. (Better viewed in PDF version)

The coefficient 2 on the right hand side of the inequality makes the result less meaningful.

B.3 A condition to have over-smoothing and over-fitting happen simultaneously

As it has alluded to before, in this paper, we argue that the performance degradation issue is mainly due to over-fitting, and over-smoothing is not likely to happen in practice. One might doubt whether over-smoothing can happen simultaneously with over-fitting. In the following, we show that over-smoothing and over-fitting might happen at the same time if a model can distinguish all nodes by only using graph structure information (e.g., node degree) and ignoring node feature information. However, such a condition is not likely in the practice.

For simplicity, let us assume that the graph contains only a single connected component. Let first consider the over-smoothing notation as defined in [40]. We can write the node embedding of node i at the ℓth layer as $\mathbf{h}_i^{(\ell)} = \mathbf{e}\mathbf{R}_{i,\ell} + \boldsymbol{\varepsilon}_{i,\ell} \in \mathbb{R}^d$, where $\mathbf{e}$ is the eigen vector associated with the largest eigenvalue, $\mathbf{R}_{i,\ell} \in \mathbb{R}^{N\times d}$ is the random projection matrix, and $\boldsymbol{\varepsilon}_{i,\ell} \in \mathbb{R}^d$ is a d-dimensional random vector such that $d_{\mathcal{M}}(\mathbf{h}_i^{(\ell)}) = \|\boldsymbol{\varepsilon}_{i,\ell}\|_2$ corresponds to its distance to subspace $\mathcal{M}$. According to the notation in [40], we have $\|\boldsymbol{\varepsilon}_{i,\ell}\|_2 \to 0$ as $\ell \to +\infty$ for any $i \in \mathcal{V}$. The minimum training loss can be achieved by choosing the final layer weight vector $\mathbf{w} \in \mathbb{R}^d$ such that $\sum_{i\in\mathcal{V}} \mathrm{Loss}\big(\mathbf{w}^\top(\mathbf{e}\mathbf{R}_{i,\ell} + \boldsymbol{\varepsilon}_{i,\ell}), y_i\big)$ is small. A small training error is achieved when $\mathbf{e}\mathbf{R}_{i,\ell}$ is discriminative. In other words, a model is under both over-smoothing and over-fitting conditions only if it can make predictions based on the graph structure, without leveraging node feature information. On the other hand, considering the over-smoothing notation as defined in [4], we know that over-smoothing happens if $\frac{\mathbf{h}_i^{(\ell)}}{\sqrt{\deg(i)}} \approx \frac{\mathbf{h}_j^{(\ell)}}{\sqrt{\deg(j)}}$ for any $i, j \in \mathcal{V}$, i.e., a classifier can distinguish any nodes based on its node degree (only based on graph information without leveraging its feature information).

C Proof of Theorem 1

Given a L-layer computation tree $\mathcal{T}^L$, let v_i^ℓ denote the ith node in the ℓth layer of a L-layer computation tree $\mathcal{T}^L$. Let suppose each node has at least d neighbors and each node has binary feature.

Let $P(v_i^\ell)$ denote the parent of node v_i^ℓ and $C(v_i^\ell)$ denote the set of children of node v_i^ℓ with $|C(v_i^\ell)| \geq d$. By the definition of computation tree, we know that $P(v_i^\ell) \in C(v_i^\ell)$ for any $\ell > 1$. As a result, we know that $C(v_i^\ell)$ has at least $(d-1)$ different choices since one of the node in $C(v_i^\ell)$ must be the same as $P(v_i^\ell)$.

Therefore, we know that a L-layer computation tree $\mathcal{T}^L$ has at least $2(d-1)^{L-1}$ different choices, where the constant 2 is because each root node has at least two different choices.

D Proof of Theorem 2

In this section, we study the convergence of L-layer deep GCN model, and show that a deeper model requires more iterations T to achieve ϵ training error. Before preceding, let first introduce the notations used for the convergence analysis of deep GCN. Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ as the graph of $N = |\mathcal{V}|$ nodes and each node in the graph is associated with a node feature and target label pair $(\mathbf{x}_i, \mathbf{y}_i)$, where $\mathbf{x}_i \in \mathbb{R}^{d_0}$ and $\mathbf{y}_i \in \mathbb{R}^{d_{L+1}}$. Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N \in \mathbb{R}^{N \times d_0}$ and $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^N \in \mathbb{R}^{N \times d_{L+1}}$ be the stack of node vector and target vector of the training data set of size N. Given a set of L matrices $\boldsymbol{\theta} = \{\mathbf{W}^{(\ell)}\}_{\ell=1}^L$, we consider the following training objective with squared loss function to learn the parameters of the model:

$$\underset{\boldsymbol{\theta}}{\text{minimize}}\ \mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2}\|\widehat{\mathbf{Y}}(\boldsymbol{\theta}) - \mathbf{Y}\|_{\mathrm{F}}^2,\ \widehat{\mathbf{Y}}(\boldsymbol{\theta}) = \mathbf{H}^{(L)}\mathbf{W}^{(L+1)},\ \mathbf{H}^{(\ell)} = \sigma(\mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)}), \tag{15}$$

where $\sigma(\cdot)$ is ReLU activation.

D.1 Useful lemmas

The following lemma analyzes the Lipschitz continuity of deep GCNs, which characterizes the change of inner layer representations by slight change of the weight parameters.

Lemma 2. *Let $\boldsymbol{\theta}_1 = \{\mathbf{W}_1^{(\ell)}\}_{\ell \in [L+1]}$ and $\boldsymbol{\theta}_2 = \{\mathbf{W}_2^{(\ell)}\}_{\ell \in [L+1]}$ be two set of parameters. Let $\lambda_\ell \geq \max\{\|\mathbf{W}_1^{(\ell)}\|_2, \|\mathbf{W}_2^{(\ell)}\|_2\}$, $s \geq \|\mathbf{L}\|_2$, and $\lambda_{i \to j} = \prod_{\ell=i}^j \lambda_\ell$, we have*

$$\|\mathbf{H}_1^{(\ell)} - \mathbf{H}_2^{(\ell)}\|_{\mathrm{F}} \leq s^\ell \lambda_{1 \to \ell} \|\mathbf{X}\|_{\mathrm{F}} \sum_{j=1}^{\ell} \lambda_j^{-1} \|\mathbf{W}_1^{(j)} - \mathbf{W}_2^{(j)}\|_2. \tag{16}$$

Proof of Lemma 2. Our proof relies on the following inequality

$$\forall \mathbf{A} \in \mathbb{R}^{a \times b}, \mathbf{B} \in \mathbb{R}^{b \times c},\ \|\mathbf{A}\mathbf{B}\|_{\mathrm{F}} \leq \|\mathbf{A}\|_2 \|\mathbf{B}\|_{\mathrm{F}} \text{ and } \|\mathbf{A}\mathbf{B}\|_{\mathrm{F}} \leq \|\mathbf{A}\|_{\mathrm{F}} \|\mathbf{B}\|_2. \tag{17}$$

By definition of $\mathbf{H}^{(\ell)}$, we have for $\ell \in [L]$

$$\begin{aligned}
\|\mathbf{H}^{(\ell)}\|_{\mathrm{F}} &= \|\sigma(\mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)})\|_{\mathrm{F}} \\
&\leq \|\mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)}\|_{\mathrm{F}} \\
&\leq \|\mathbf{L}\|_2 \|\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)}\|_{\mathrm{F}} \\
&\leq \|\mathbf{L}\|_2 \|\mathbf{W}^{(\ell)}\|_2 \|\mathbf{H}^{(\ell-1)}\|_{\mathrm{F}} \\
&\leq \|\mathbf{X}\|_{\mathrm{F}} \prod_{j=1}^{\ell} \Big(\|\mathbf{L}\|_2 \|\mathbf{W}^{(\ell)}\|_2\Big).
\end{aligned} \tag{18}$$

Let $\boldsymbol{\theta}_1 = \{\mathbf{W}_1^{(\ell)}\}_{\ell\in[L+1]}$ and $\boldsymbol{\theta}_2 = \{\mathbf{W}_2^{(\ell)}\}_{\ell\in[L+1]}$ be two set of parameters, we have

$$
\begin{aligned}
\|\mathbf{H}_1^{(\ell)} - \mathbf{H}_2^{(\ell)}\|_{\mathrm{F}} &= \|\sigma(\mathbf{L}\mathbf{H}_1^{(\ell-1)}\mathbf{W}_1^{(\ell)}) - \sigma(\mathbf{L}\mathbf{H}_2^{(\ell-1)}\mathbf{W}_2^{(\ell)})\|_{\mathrm{F}} \\
&\underset{(a)}{\leq} \|\mathbf{L}\mathbf{H}_1^{(\ell-1)}\mathbf{W}_1^{(\ell)} - \mathbf{L}\mathbf{H}_2^{(\ell-1)}\mathbf{W}_2^{(\ell)}\|_{\mathrm{F}} \\
&\leq \|\mathbf{L}\|_2\|\mathbf{H}_1^{(\ell-1)}\mathbf{W}_1^{(\ell)} - \mathbf{H}_2^{(\ell-1)}\mathbf{W}_2^{(\ell)}\|_{\mathrm{F}} \\
&= \|\mathbf{L}\|_2\left(\|\mathbf{H}_1^{(\ell-1)}(\mathbf{W}_1^{(\ell)} - \mathbf{W}_2^{(\ell)}) + (\mathbf{H}_1^{(\ell-1)} - \mathbf{H}_2^{(\ell-1)})\mathbf{W}_2^{(\ell)}\|_{\mathrm{F}}\right) \\
&\leq \|\mathbf{L}\|_2\left(\|\mathbf{H}_1^{(\ell-1)}\|_{\mathrm{F}}\|\mathbf{W}_1^{(\ell)} - \mathbf{W}_2^{(\ell)}\|_2 + \|\mathbf{H}_1^{(\ell-1)} - \mathbf{H}_2^{(\ell-1)}\|_{\mathrm{F}}\|\mathbf{W}_2^{(\ell)}\|_2\right) \\
&\underset{(b)}{\leq} \|\mathbf{L}\|_2\|\mathbf{X}\|_{\mathrm{F}}\left[\prod_{j=1}^{\ell-1}\left(\|\mathbf{L}\|_2\|\mathbf{W}_1^{(\ell)}\|_2\right)\right]\|\mathbf{W}_1^{(\ell)} - \mathbf{W}_2^{(\ell)}\|_2 \\
&\quad + \|\mathbf{L}\|_2\|\mathbf{W}_2^{(\ell)}\|_2\|\mathbf{H}_1^{(\ell-1)} - \mathbf{H}_2^{(\ell-1)}\|_{\mathrm{F}},
\end{aligned}
\tag{19}
$$

where (a) is due to $\sigma(\cdot)$ is 1-Lipschitz continuous, (b) is due to Eq. 18.

Let $\lambda_\ell \geq \max\{\|\mathbf{W}_1^{(\ell)}\|_2, \|\mathbf{W}_2^{(\ell)}\|_2\}$, $s \geq \|\mathbf{L}\|_2$, and $\lambda_{i\to j} = \prod_{\ell=i}^{j}\lambda_\ell$, , we have

$$
\begin{aligned}
\|\mathbf{H}_1^{(\ell)} - \mathbf{H}_2^{(\ell)}\|_{\mathrm{F}} &\leq s^\ell\lambda_{1\to(\ell-1)}\|\mathbf{X}\|_{\mathrm{F}}\|\mathbf{W}_1^{(\ell)} - \mathbf{W}_2^{(\ell)}\|_2 + s\lambda_\ell\|\mathbf{H}_1^{(\ell-1)} - \mathbf{H}_2^{(\ell-1)}\|_{\mathrm{F}} \\
&\leq s^\ell\lambda_{1\to\ell}\|\mathbf{X}\|_{\mathrm{F}}\left(\sum_{j=1}^{\ell}\lambda_j^{-1}\|\mathbf{W}_1^{(j)} - \mathbf{W}_2^{(j)}\|_2\right).
\end{aligned}
\tag{20}
$$

□

The following lemma derives the upper bound of the gradient with respect to the weight parameters, which plays an important role in the convergence analysis of deep GCN.

Lemma 3. *Let $\boldsymbol{\theta} = \{\mathbf{W}^{(\ell)}\}_{\ell\in[L+1]}$ be the set of parameters, $\lambda_\ell \geq \max\{\|\mathbf{W}_1^{(\ell)}\|_2, \|\mathbf{W}_2^{(\ell)}\|_2\}$, $s \geq \|\mathbf{L}\|_2$, and $\lambda_{i\to j} = \prod_{\ell=i}^{j}\lambda_\ell$, we have*

$$
\left\|\frac{\partial\mathcal{L}(\boldsymbol{\theta})}{\partial\mathbf{W}^{(\ell)}}\right\|_{\mathrm{F}} \leq \frac{s^L\lambda_{1\to(L+1)}}{\lambda_\ell}\|\mathbf{X}\|_{\mathrm{F}}\|\widehat{\mathbf{Y}} - \mathbf{Y}\|_{\mathrm{F}}.
\tag{21}
$$

Proof of Lemma 3. By the definition of $\frac{\partial\mathcal{L}(\boldsymbol{\theta})}{\partial\mathbf{W}^{(\ell)}}$, we have

$$
\begin{aligned}
\left\|\frac{\partial\mathcal{L}(\boldsymbol{\theta})}{\partial\mathbf{W}^{(\ell)}}\right\|_{\mathrm{F}} &= \left\|\left(\frac{\partial\widehat{\mathbf{Y}}}{\partial\mathbf{W}^{(\ell)}}\right)^\top(\widehat{\mathbf{Y}} - \mathbf{Y})\right\|_{\mathrm{F}} \\
&= \left\|\frac{\partial[\mathbf{I}_N\otimes(\mathbf{W}^{(L+1)})^\top]\mathrm{vec}(\mathbf{H}^{(L)})}{\partial\mathrm{vec}(\mathbf{W}^{(\ell)})}\mathrm{vec}(\widehat{\mathbf{Y}} - \mathbf{Y})\right\|_2 \\
&= \left\|[\mathbf{I}_N\otimes(\mathbf{W}^{(L+1)})^\top]\frac{\partial\mathrm{vec}(\mathbf{H}^{(L)})}{\partial\mathrm{vec}(\mathbf{H}^{(\ell)})}\frac{\partial\mathrm{vec}(\mathbf{H}^{(\ell)})}{\partial\mathrm{vec}(\mathbf{W}^{(\ell)})}\mathrm{vec}(\widehat{\mathbf{Y}} - \mathbf{Y})\right\|_2 \\
&\underset{(a)}{\leq} \left\|[\mathbf{I}_N\otimes(\mathbf{W}^{(L+1)})^\top]\frac{\partial\mathrm{vec}(\mathbf{L}\mathbf{H}^{(L-1)}\mathbf{W}^{(L)})}{\partial\mathrm{vec}(\mathbf{H}^{(\ell)})}\frac{\partial\mathrm{vec}(\mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)})}{\partial\mathrm{vec}(\mathbf{W}^{(\ell)})}\mathrm{vec}(\widehat{\mathbf{Y}} - \mathbf{Y})\right\|_2 \\
&= \left\|[\mathbf{I}_N\otimes(\mathbf{W}^{(L+1)})^\top]\frac{\partial\mathrm{vec}(\mathbf{L}\otimes(\mathbf{W}^{(L)})^\top)\mathrm{vec}(\mathbf{H}^{(L-1)})}{\partial\mathrm{vec}(\mathbf{H}^{(\ell)})}\frac{\partial[\mathbf{L}\mathbf{H}^{(\ell-1)}\otimes\mathbf{I}_{d_\ell}]\mathrm{vec}(\mathbf{W}^{(\ell)})}{\partial\mathrm{vec}(\mathbf{W}^{(\ell)})}\mathrm{vec}(\widehat{\mathbf{Y}} - \mathbf{Y})\right\|_2 \\
&= \left\|[\mathbf{I}_N\otimes(\mathbf{W}^{(L+1)})^\top]\mathrm{vec}\left(\mathbf{L}^{L-\ell}\otimes(\mathbf{W}^{(\ell+1)}\dots\mathbf{W}^{(L)})^\top\right)[\mathbf{L}\mathbf{H}^{(\ell-1)}\otimes\mathbf{I}_{d_\ell}]\mathrm{vec}(\widehat{\mathbf{Y}} - \mathbf{Y})\right\|_2 \\
&= \left\|\underbrace{(\mathbf{L}^{L-\ell+1}\mathbf{H}^{(\ell-1)})\otimes(\mathbf{W}^{(\ell+1)}\dots\mathbf{W}^{(L+1)})^\top}_{\mathbb{R}^{Nd_{L+1}\times d_{\ell-1}d_\ell}}\mathrm{vec}(\widehat{\mathbf{Y}} - \mathbf{Y})\right\|_{\mathrm{F}}
\end{aligned}
$$

$$= \left\| (\mathbf{W}^{(\ell+1)} \dots \mathbf{W}^{(L+1)})(\widehat{\mathbf{Y}} - \mathbf{Y})^\top \mathbf{L}^{L-\ell+1} \mathbf{H}^{(\ell-1)} \right\|_2$$
$$\leq s^{L-\ell+1} \lambda_{(\ell+1)\to(L+1)} \|\widehat{\mathbf{Y}} - \mathbf{Y}\|_{\mathrm{F}} \|\mathbf{H}^{(\ell-1)}\|_{\mathrm{F}},$$

where (a) is due to $\|\sigma(x)\|_2 \leq \|x\|_2$.

Using similar proof strategy, we can upper bound $\|\mathbf{H}^{(\ell-1)}\|_{\mathrm{F}}$ by

$$\begin{aligned} \|\mathbf{H}^{(\ell-1)}\|_{\mathrm{F}} &= \|\sigma(\mathbf{L}\mathbf{H}^{(\ell-2)}\mathbf{W}^{(\ell-1)})\|_{\mathrm{F}} \\ &\leq \|\mathbf{L}\mathbf{H}^{(\ell-2)}\mathbf{W}^{(\ell-1)}\|_{\mathrm{F}} \\ &\leq s^{\ell-1}\lambda_{1\to(\ell-1)}\|\mathbf{X}\|_{\mathrm{F}}. \end{aligned} \tag{22}$$

By combining the above two equations, we have

$$\left\| \frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \mathbf{W}^{(\ell)}} \right\|_{\mathrm{F}} \leq \frac{s^L \lambda_{1\to(L+1)}}{\lambda_\ell} \|\mathbf{X}\|_{\mathrm{F}} \|\widehat{\mathbf{Y}} - \mathbf{Y}\|_{\mathrm{F}}. \tag{23}$$

□

D.2 Main result

In the following, we provide the convergence analysis on deep GCNs. In particular, Theorem 7 shows that under assumption on the weight initialization (Eq. 24) and the width of the last layer's input dimension ($d_L \geq N$), the deep GCN enjoys linear convergence rate. Recall our discussion in Section 4 that $\lambda_\ell \geq 1$ (by Gordon's Theorem for Gaussian matrices) and $\|\mathbf{L}\|_2 = 1$ for $\mathbf{L} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, we know that $s\lambda_\ell \geq 1$, thus deeper model requires a smaller learning rate η according to Eq. 25. Since the number of training iteration T is reverse proportional to learning rate η, which explains why a deeper model requires more training iterations than the shallow one.

Theorem 7. *Consider a deep GCN with ReLU activation (defined in Eq. 15) where the width of the last hidden layer satisfies $d_L \geq N$. Let $\{C_\ell\}_{\ell\in[L+1]}$ be any sequence of positive numbers and define $\alpha_0 = \lambda_{\min}(\mathbf{H}_0^{(L)})$, $\lambda_\ell = \|\mathbf{W}_0^{(\ell)}\|_2 + C_\ell$, $\lambda_{i\to j} = \prod_{\ell=i}^{j} \lambda_\ell$.*

Assume the following conditions are satisfied at the initialization:

$$\begin{aligned} \alpha_0^2 &\geq 16 s^{2L} \|\mathbf{X}\|_{\mathrm{F}} \max_{\ell\in[L+1]} \frac{\lambda_{1\to(L+1)}}{\lambda_\ell C_\ell} \sqrt{2\mathcal{L}(\boldsymbol{\theta}_0)}, \\ \alpha_0^3 &\geq 32 s^{2L} \|\mathbf{X}\|_{\mathrm{F}}^2 \lambda_{L+1} \sum_{\ell=1}^{L} \frac{\lambda_{1\to L}^2}{\lambda_\ell^2} \sqrt{2\mathcal{L}(\boldsymbol{\theta}_0)}, \\ \alpha_0^2 &\geq 16 s^{2L} \|\mathbf{X}\|_{\mathrm{F}}^2 \lambda_{L+1}^2 \sum_{\ell=1}^{L} \frac{\lambda_{1\to L}^2}{\lambda_\ell^2}. \end{aligned} \tag{24}$$

Let the learning rate satisfies

$$\eta \leq \min\left(\frac{8}{\alpha_0^2}, \frac{\left(\sum_{\ell=1}^{L} \lambda_\ell^{-2}\right)}{s^{2L}\lambda_{1\to(L+1)}^2 \|\mathbf{X}\|_{\mathrm{F}}^2 \left(\sum_{\ell=1}^{L+1} \lambda_\ell^{-2}\right)^2} \right). \tag{25}$$

Then deep GCN can achieve $\mathcal{L}(\boldsymbol{\theta}_T) \leq \epsilon$ for any T satisfied

$$T \geq \frac{8}{\eta\alpha_0^2} \log\left(\frac{\mathcal{L}(\boldsymbol{\theta}_0)}{\epsilon}\right). \tag{26}$$

Proof of Theorem 7. The proof follows the proof of Theorem 2.2 in [39] by extending result from MLP to GCN. The key idea of the proof is to show the followings hold by induction:

$$\begin{cases} \|\mathbf{W}_r^{(\ell)}\|_2 \leq \lambda_\ell & \text{for all } \ell \in [L],\ r \in [0,t], \\ \lambda_{\min}(\mathbf{H}_r^{(L)}) \geq \frac{\alpha_0}{2} & \text{for all } r \in [0,t], \\ \mathcal{L}(\boldsymbol{\theta}_r) \leq (1 - \eta\alpha_0^2/8)\mathcal{L}(\boldsymbol{\theta}_0) & \text{for all } r \in [0,t]. \end{cases} \tag{27}$$

Clearly, Eq. 27 holds for $t = 0$. Let assume Eq. 27 holds up to iteration t, and let show it also holds for $(t+1)$th iteration.

Let first prove the first inequality holds in Eq. 27. By the triangle inequality, we can decompose the distance between $\mathbf{W}_{t+1}^{(\ell)}$ to $\mathbf{W}_0^{(\ell)}$ by

$$\begin{aligned}
\|\mathbf{W}_{t+1}^{(\ell)} - \mathbf{W}_0^{(\ell)}\|_{\mathrm{F}} &\leq \sum_{k=0}^{t} \|\mathbf{W}_{k+1}^{(\ell)} - \mathbf{W}_k^{(\ell)}\|_{\mathrm{F}} \\
&= \eta \sum_{k=0}^{t} \left\| \frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \mathbf{W}_k^{(\ell)}} \right\|_{\mathrm{F}} \\
&\leq \eta s^L \lambda_{1\to(L+1)} \lambda_\ell^{-1} \|\mathbf{X}\|_{\mathrm{F}} \sum_{k=0}^{t} \|\widehat{\mathbf{Y}}_k - \mathbf{Y}\|_{\mathrm{F}} \\
&\leq \eta s^L \lambda_{1\to(L+1)} \lambda_\ell^{-1} \|\mathbf{X}\|_{\mathrm{F}} \sum_{k=0}^{t} \sqrt{2\mathcal{L}(\boldsymbol{\theta}_k)} \\
&\underset{(a)}{\leq} \eta s^L \lambda_{1\to(L+1)} \lambda_\ell^{-1} \|\mathbf{X}\|_{\mathrm{F}} \sum_{k=0}^{t} \left(1 - \eta \frac{\alpha_0^2}{8}\right)^{k/2} \sqrt{2\mathcal{L}(\boldsymbol{\theta}_0)},
\end{aligned} \tag{28}$$

where inequality (a) follows the induction condition.

Let $u = \sqrt{1 - \eta\alpha_0^2/8}$ we have $\eta = 8(1-u^2)/\alpha_0^2$ and

$$\sum_{k=0}^{t} \left(1 - \eta \frac{\alpha_0^2}{8}\right)^{k/2} = \sum_{k=0}^{t} u^k = \frac{1-u^{t+1}}{1-u}. \tag{29}$$

Then, plugging the result into Eq. 28, we have

$$\begin{aligned}
\|\mathbf{W}_{t+1}^{(\ell)} - \mathbf{W}_0^{(\ell)}\|_{\mathrm{F}} &\leq s^L \lambda_{1\to(L+1)} \lambda_\ell^{-1} \|\mathbf{X}\|_{\mathrm{F}} \frac{8(1-u^2)}{\alpha_0^2} \frac{1-u^{t+1}}{1-u} \sqrt{2\mathcal{L}(\boldsymbol{\theta}_0)}, \\
&\underset{(a)}{\leq} \frac{16}{\alpha_0^2} s^L \lambda_{1\to(L+1)} \lambda_\ell^{-1} \|\mathbf{X}\|_{\mathrm{F}} \sqrt{2\mathcal{L}(\boldsymbol{\theta}_0)},
\end{aligned} \tag{30}$$

where (a) is due to $u \in (0,1)$. Let C_ℓ as a positive constant that

$$C_\ell \geq \frac{16}{\alpha_0^2} s^L \lambda_{1\to(L+1)} \lambda_\ell^{-1} \|\mathbf{X}\|_{\mathrm{F}} \sqrt{2\mathcal{L}(\boldsymbol{\theta}_0)}. \tag{31}$$

By Wely's inequality, we have

$$\|\mathbf{W}_{t+1}^{(\ell)}\|_2 \leq \|\mathbf{W}_0^{(\ell)}\|_2 + C_\ell = \lambda_\ell. \tag{32}$$

To prove the second inequality holds in Eq. 27, we first upper bound $\|\mathbf{H}_{t+1}^{(L)} - \mathbf{H}_0^{(L)}\|_{\mathrm{F}}$ by

$$\begin{aligned}
\|\mathbf{H}_{t+1}^{(L)} - \mathbf{H}_0^{(L)}\|_{\mathrm{F}} &\underset{(a)}{\leq} s^L \lambda_{1\to L} \|\mathbf{X}\|_{\mathrm{F}} \sum_{\ell=1}^{L} \lambda_\ell^{-1} \|\mathbf{W}_{t+1}^{(\ell)} - \mathbf{W}_0^{(\ell)}\|_2 \\
&\underset{(b)}{\leq} s^L \lambda_{1\to L} \|\mathbf{X}\|_{\mathrm{F}} \sum_{\ell=1}^{L} \lambda_\ell^{-1} \left(\frac{16}{\alpha_0^2} s^L \lambda_{1\to(L+1)} \lambda_\ell^{-1} \|\mathbf{X}\|_{\mathrm{F}} \sqrt{2\mathcal{L}(\boldsymbol{\theta}_0)} \right) \\
&= \frac{16}{\alpha_0^2} s^{2L} \lambda_{1\to L} \lambda_{1\to(L+1)} \|\mathbf{X}\|_{\mathrm{F}}^2 \sqrt{2\mathcal{L}(\boldsymbol{\theta}_0)} \left(\sum_{\ell=1}^{L} \lambda_\ell^{-2} \right) \\
&\underset{(c)}{\leq} \frac{\alpha_0}{2},
\end{aligned} \tag{33}$$

where (a) is due to Lemma 2, (b) is due to Eq. 30, and (c) is due to the second condition in Eq. 24.

By Weyl's inequality, this implies $|\lambda_{\min}(\mathbf{H}_{t+1}^{(L)}) - \lambda_{\min}(\mathbf{H}_0^{(L)})| = |\lambda_{\min}(\mathbf{H}_{t+1}^{(L)}) - \alpha_0| \leq \frac{\alpha_0}{2}$ and

$$\lambda_{\min}(\mathbf{H}_{t+1}^{(L)}) \geq \frac{\alpha_0}{2}. \tag{34}$$

Let $\mathbf{G} = \mathbf{H}_t^{(L)}\mathbf{W}_{t+1}^{(L+1)}$, then we have

$$\begin{aligned}
2\mathcal{L}(\boldsymbol{\theta}_{t+1}) &= \|\widehat{\mathbf{Y}}_{t+1} - \mathbf{Y}\|_{\mathrm{F}}^2 \\
&= \|\widehat{\mathbf{Y}}_{t+1} - \widehat{\mathbf{Y}}_t + \widehat{\mathbf{Y}}_t - \mathbf{Y}\|_{\mathrm{F}}^2 \\
&= \|\widehat{\mathbf{Y}}_t - \mathbf{Y}\|_{\mathrm{F}}^2 + \|\widehat{\mathbf{Y}}_{t+1} - \widehat{\mathbf{Y}}_t\|_{\mathrm{F}}^2 + 2\langle \widehat{\mathbf{Y}}_t - \mathbf{Y}, \widehat{\mathbf{Y}}_{t+1} - \widehat{\mathbf{Y}}_t\rangle_{\mathrm{F}} \\
&= 2\mathcal{L}(\boldsymbol{\theta}_t) + \|\widehat{\mathbf{Y}}_{t+1} - \widehat{\mathbf{Y}}_t\|_{\mathrm{F}}^2 + 2\langle \widehat{\mathbf{Y}}_t - \mathbf{Y}, \widehat{\mathbf{Y}}_{t+1} - \mathbf{G}\rangle_{\mathrm{F}} + 2\langle \widehat{\mathbf{Y}}_t - \mathbf{Y}, \mathbf{G} - \widehat{\mathbf{Y}}_t\rangle_{\mathrm{F}}.
\end{aligned} \tag{35}$$

We can upper bound $\|\widehat{\mathbf{Y}}_{t+1} - \widehat{\mathbf{Y}}_t\|_{\mathrm{F}}^2$ in Eq. 35 by

$$\begin{aligned}
\|\widehat{\mathbf{Y}}_{t+1} - \widehat{\mathbf{Y}}_t\|_{\mathrm{F}} &= \|\mathbf{H}_{t+1}^{(L)}\mathbf{W}_{t+1}^{(L+1)} - \mathbf{H}_t^{(L)}\mathbf{W}_t^{(L+1)}\|_{\mathrm{F}} \\
&= \|\mathbf{H}_{t+1}^{(L)}(\mathbf{W}_{t+1}^{(L+1)} - \mathbf{W}_t^{(L+1)}) + (\mathbf{H}_{t+1}^{(L)} - \mathbf{H}_t^{(L)})\mathbf{W}_t^{(L+1)}\|_{\mathrm{F}} \\
&\leq \|\mathbf{H}_{t+1}^{(L)}\|_{\mathrm{F}}\|\mathbf{W}_{t+1}^{(L+1)} - \mathbf{W}_t^{(L+1)}\|_2 + \|\mathbf{H}_{t+1}^{(L)} - \mathbf{H}_t^{(L)}\|_{\mathrm{F}}\|\mathbf{W}_t^{(L+1)}\|_2 \\
&\underset{(a)}{\leq} s^L\lambda_{1\to(L+1)}\|\mathbf{X}\|_{\mathrm{F}}\sum_{\ell=1}^{L}\lambda_\ell^{-1}\|\mathbf{W}_{t+1}^{(\ell)} - \mathbf{W}_t^{(\ell)}\|_2 + s^L\lambda_{1\to L}\|\mathbf{X}\|_{\mathrm{F}}\|\mathbf{W}_{t+1}^{(L+1)} - \mathbf{W}_t^{(L+1)}\|_2 \\
&= s^L\lambda_{1\to(L+1)}\|\mathbf{X}\|_{\mathrm{F}}\sum_{\ell=1}^{L+1}\lambda_\ell^{-1}\|\mathbf{W}_{t+1}^{(\ell)} - \mathbf{W}_t^{(\ell)}\|_2 \\
&\underset{(b)}{\leq} \eta s^L\lambda_{1\to(L+1)}\|\mathbf{X}\|_{\mathrm{F}}\sum_{\ell=1}^{L+1}\lambda_\ell^{-1}\left(\frac{s^L\lambda_{1\to(L+1)}}{\lambda_\ell}\|\mathbf{X}\|_{\mathrm{F}}\|\widehat{\mathbf{Y}}_t - \mathbf{Y}\|_{\mathrm{F}}\right) \\
&= \eta s^{2L}\lambda_{1\to(L+1)}^2\|\mathbf{X}\|_{\mathrm{F}}^2\left(\sum_{\ell=1}^{L+1}\lambda_\ell^{-2}\right)\|\widehat{\mathbf{Y}}_t - \mathbf{Y}\|_{\mathrm{F}},
\end{aligned} \tag{36}$$

where (a) is due to Eq. 18 and Lemma 2, and (b) follows from Lemma 3.

We can upper bound $\langle \widehat{\mathbf{Y}}_t - \mathbf{Y}, \widehat{\mathbf{Y}}_{t+1} - \mathbf{G}\rangle_{\mathrm{F}}$ in Eq. 35 by

$$\begin{aligned}
&\langle \widehat{\mathbf{Y}}_t - \mathbf{Y}, \widehat{\mathbf{Y}}_{t+1} - \mathbf{G}\rangle_{\mathrm{F}} \\
&\leq \|\widehat{\mathbf{Y}}_t - \mathbf{Y}\|_{\mathrm{F}}\|\widehat{\mathbf{Y}}_{t+1} - \mathbf{G}\|_{\mathrm{F}} \\
&= \|\widehat{\mathbf{Y}}_t - \mathbf{Y}\|_{\mathrm{F}}\|(\mathbf{H}_{t+1}^{(L)} - \mathbf{H}_t^{(L)})\mathbf{W}_{t+1}^{(L+1)}\|_{\mathrm{F}} \\
&\underset{(a)}{\leq} \lambda_{L+1}\|\widehat{\mathbf{Y}}_t - \mathbf{Y}\|_{\mathrm{F}}\left(s^L\lambda_{1\to L}\|\mathbf{X}\|_{\mathrm{F}}\sum_{\ell=1}^{L}\lambda_\ell^{-1}\|\mathbf{W}_{t+1}^{(\ell)} - \mathbf{W}_t^{(\ell)}\|_2\right) \\
&= s^L\lambda_{1\to(L+1)}\|\widehat{\mathbf{Y}}_t - \mathbf{Y}\|_{\mathrm{F}}\|\mathbf{X}\|_{\mathrm{F}}\left(\sum_{\ell=1}^{L}\lambda_\ell^{-1}\|\mathbf{W}_{t+1}^{(\ell)} - \mathbf{W}_t^{(\ell)}\|_2\right) \\
&\underset{(b)}{\leq} \eta s^L\lambda_{1\to(L+1)}\|\widehat{\mathbf{Y}}_t - \mathbf{Y}\|_{\mathrm{F}}\|\mathbf{X}\|_{\mathrm{F}}\left(\sum_{\ell=1}^{L}\lambda_\ell^{-1}\frac{s^L\lambda_{1\to(L+1)}}{\lambda_\ell}\|\mathbf{X}\|_{\mathrm{F}}\|\widehat{\mathbf{Y}} - \mathbf{Y}\|_{\mathrm{F}}\right) \\
&= \eta s^{2L}\lambda_{1\to(L+1)}^2\|\widehat{\mathbf{Y}}_t - \mathbf{Y}\|_{\mathrm{F}}^2\|\mathbf{X}\|_{\mathrm{F}}^2\left(\sum_{\ell=1}^{L}\lambda_\ell^{-2}\right),
\end{aligned} \tag{37}$$

where (a) is due to Lemma 2, and (b) follows from Lemma 3.

We can upper bound $\langle \widehat{\mathbf{Y}}_t - \mathbf{Y}, \mathbf{G} - \widehat{\mathbf{Y}}_t \rangle_{\mathrm{F}}$ by

$$
\begin{aligned}
\langle \widehat{\mathbf{Y}}_t - \mathbf{Y}, \mathbf{G} - \widehat{\mathbf{Y}}_t \rangle_{\mathrm{F}} &= \langle \widehat{\mathbf{Y}}_t - \mathbf{Y}, \mathbf{H}_t^{(L)} (\mathbf{W}_{t+1}^{(L+1)} - \mathbf{W}_t^{(L+1)}) \rangle_{\mathrm{F}} \\
&= \eta \left\langle \widehat{\mathbf{Y}}_t - \mathbf{Y}, \mathbf{H}_t^{(L)} \frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \mathbf{W}_t^{(L+1)}} \right\rangle_{\mathrm{F}} \\
&= -\eta \left\langle \widehat{\mathbf{Y}}_t - \mathbf{Y}, \mathbf{H}_t^{(L)} \left((\mathbf{H}_t^{(L)})^\top (\widehat{\mathbf{Y}}_t - \mathbf{Y}) \right) \right\rangle_{\mathrm{F}} \\
&\underset{(a)}{=} -\eta \operatorname{tr} \left((\widehat{\mathbf{Y}}_t - \mathbf{Y})^\top \mathbf{H}_t^{(L)} (\mathbf{H}_t^{(L)})^\top (\widehat{\mathbf{Y}}_t - \mathbf{Y}) \right) \\
&\underset{(b)}{=} -\eta \operatorname{tr} \left(\mathbf{H}_t^{(L)} (\mathbf{H}_t^{(L)})^\top (\widehat{\mathbf{Y}}_t - \mathbf{Y}) (\widehat{\mathbf{Y}}_t - \mathbf{Y})^\top \right) \\
&\underset{(c)}{\leq} -\eta \lambda_{\min} \left(\mathbf{H}_t^{(L)} (\mathbf{H}_t^{(L)})^\top \right) \operatorname{tr} \left((\widehat{\mathbf{Y}}_t - \mathbf{Y}) (\widehat{\mathbf{Y}}_t - \mathbf{Y})^\top \right) \\
&\underset{(d)}{=} -\eta \lambda_{\min} \left(\mathbf{H}_t^{(L)} (\mathbf{H}_t^{(L)})^\top \right) \| \widehat{\mathbf{Y}}_t - \mathbf{Y} \|_{\mathrm{F}}^2 \\
&\underset{(e)}{=} -\eta \lambda_{\min}^2 \left(\mathbf{H}_t^{(L)} \right) \| \widehat{\mathbf{Y}}_t - \mathbf{Y} \|_{\mathrm{F}}^2, \\
&\underset{(f)}{\leq} -\eta \frac{\alpha_0^2}{4} \| \widehat{\mathbf{Y}}_t - \mathbf{Y} \|_{\mathrm{F}}^2,
\end{aligned}
\tag{38}
$$

where (a) is due to $\langle \mathbf{A}, \mathbf{B} \rangle = \operatorname{tr}(\mathbf{A}^\top \mathbf{B})$, (b) is because of the cyclic property of the trace, (c) is due to $\operatorname{tr}(\mathbf{AB}) \geq \lambda_{\min}(\mathbf{A}) \operatorname{tr}(\mathbf{B})$ for any symmetric matrices $\mathbf{A}, \mathbf{B} \in \mathbb{S}$ and $\mathbf{B}$ is positive semi-definite, (d) is due to $\operatorname{tr}(\mathbf{AA}^\top) = \|\mathbf{A}\|_{\mathrm{F}}^2$, (e) requires the $d_L \geq N$ assumption[8], and (f) is due to Eq. 34.

Let $A = s^{2L} \lambda_{1 \to (L+1)}^2 \|\mathbf{X}\|_{\mathrm{F}}^2 \left(\sum_{\ell=1}^{L+1} \lambda_\ell^{-2} \right)$ and $B = s^{2L} \lambda_{1 \to (L+1)}^2 \|\mathbf{X}\|_{\mathrm{F}}^2 \left(\sum_{\ell=1}^{L} \lambda_\ell^{-2} \right)$, we have

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}_{t+1}) &\leq \left(1 - \eta \frac{\alpha_0^2}{4} + \eta^2 A^2 + \eta B \right) \mathcal{L}(\boldsymbol{\theta}_t) \\
&\underset{(a)}{\leq} \left(1 - \eta \left(\frac{\alpha_0^2}{4} - 2B \right) \right) \mathcal{L}(\boldsymbol{\theta}_t) \\
&\underset{(b)}{\leq} \left(1 - \eta \frac{\alpha_0^2}{8} \right) \mathcal{L}(\boldsymbol{\theta}_t),
\end{aligned}
\tag{39}
$$

where (a) requires choosing η such that $\eta \leq B/A^2$, and (b) holds due to the first condition in Eq. 24.

Therefore, we have

$$
\mathcal{L}(\boldsymbol{\theta}_T) \leq \left(1 - \eta \frac{\alpha_0^2}{8} \right)^T \mathcal{L}(\boldsymbol{\theta}_0) \leq \exp \left(-\eta T \frac{\alpha_0^2}{8} \right) \mathcal{L}(\boldsymbol{\theta}_0) = \epsilon. \tag{40}
$$

By taking log on both side, we have

$$
T \geq \frac{8}{\eta \alpha_0^2} \log \left(\frac{\mathcal{L}(\boldsymbol{\theta}_0)}{\epsilon} \right). \tag{41}
$$

□

[8]The equality (e) follows from the fact that given a fat matrix $\mathbf{A} \in \mathbb{R}^{N \times d}$ with $d > N$, all the eigenvalues of $\mathbf{AA}^\top$ are just the square of the singular values of $\mathbf{A}$. However, note that this is no longer true when $d < N$ since in this case we have $\mathbf{AA}^\top$ being a low rank matrix, hence its smallest eigenvalue is zero, whereas the smallest singular value of $\mathbf{A}$ (i.e., the $\min(N, d)$-th singular value of $\mathbf{A}$) can be strictly positive if $\mathbf{A}$ is full rank, hence (e) does not hold.

E Additional empirical results

In this section, we report additional empirical results to illustrate the correctness of our theoretical analysis and the benefits of decoupled GCN structure.

E.1 Comparison of generalization error on the real-world datasets.

We empirically compare the generalization error of different GCN structures on the real-world datasets, where the generalization error is measured by the difference between validation loss and training loss.

Setups. We select hidden dimension as 64, $\alpha_\ell = 0.9$ for APPNP and GCNII, $\beta_\ell \approx \frac{0.5}{\ell}$ for GCNII, and without any weight decay (ℓ_2 regularization on weight matrices) or dropout operators. Note that although in our theoretical analysis we suppose β_ℓ is a constant that does not change with respect to the layers, in this experiment we follow the configuration in [6] by selecting $\beta_\ell = \log(\frac{0.5}{\ell} + 1)$, which guarantees a small generalization error on most small scale datasets. The same setup is also used for the results on synthetic dataset in Figure 4.

Results. As shown in Figure 11 and Figure 12, ResGCN has the largest generalization gap due to the skip-connections, APPNP has the smallest generalization error by removing the weight matrices in each individual layers. On the other hand, GCNII achieves a good balance between GCN and APPNP by balancing the expressive and generalization power. Finally, DGCN enjoys a small generalization error by using the decoupled GCN structure.

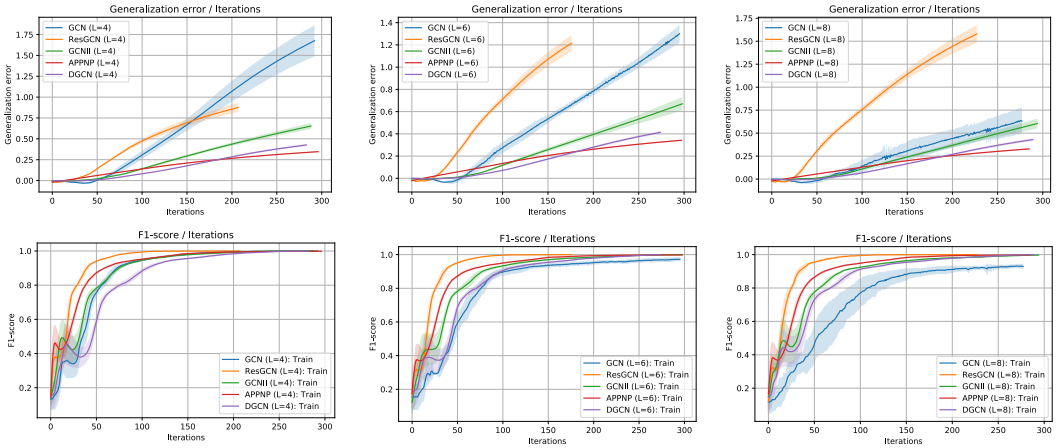

Figure 11: Comparison of generalization error and training F1-score on the *Cora* dataset. The curve stops early at the largest training accuracy iteration.

E.2 Effect of hyper-parameters in GCNII.

In the following, we first compare the effect of α_ℓ on the generalization error of GCNII. We select the hidden dimension as 64, and no dropout or weight decay is used for the training. As shown in Figure 13 and Figure 14, increasing α_ℓ leads to a smaller generalization error but a slower convergence speed, i.e., GCNII trades off the expressiveness for generalization power by increasing α_ℓ from 0 to 1. In practice, $\alpha_\ell = 0.9$ is utilized in GCNII [6] for empirical evaluations.[9]

Then, we compare the effect of β_ℓ on the generalization error of GCNII. Similar to the previous experiments, we choose hidden dimension as 64, without applying dropout or weight decay during training. As shown in Figure 15 and Figure 16, decreasing β_ℓ leads to a smaller generalization error. In practice, $\beta_\ell = \log(\frac{0.5}{\ell} + 1)$ is utilized in GCNII [6] for empirical evaluations, which guarantees a small generalization error.

[9] Note that the definition of α_ℓ we use in this paper is different from the definition in [6], where $\alpha_\ell = 0.9$ in this paper stands for the selection of alpha as $1 - \alpha_\ell = 0.1$ in [6].

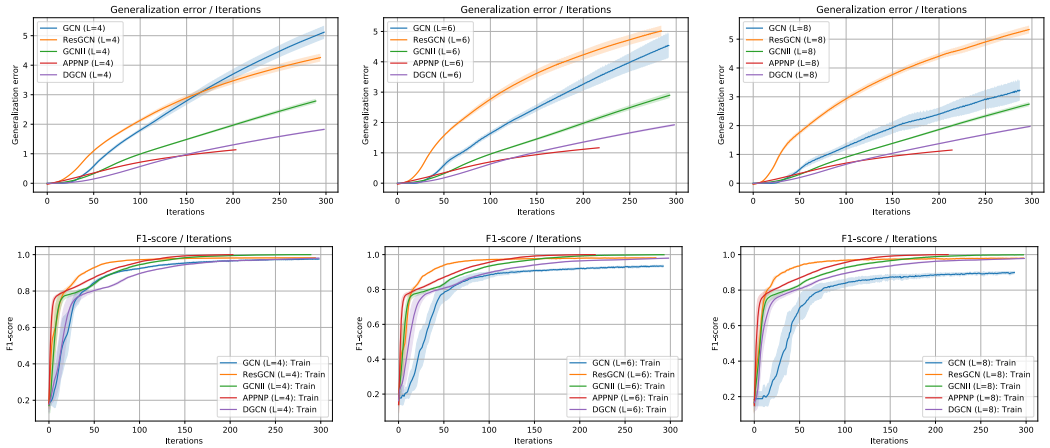

Figure 12: Comparison of generalization error and training F1-score on the *Citeseer* dataset. The curve stops early at the largest training accuracy iteration.

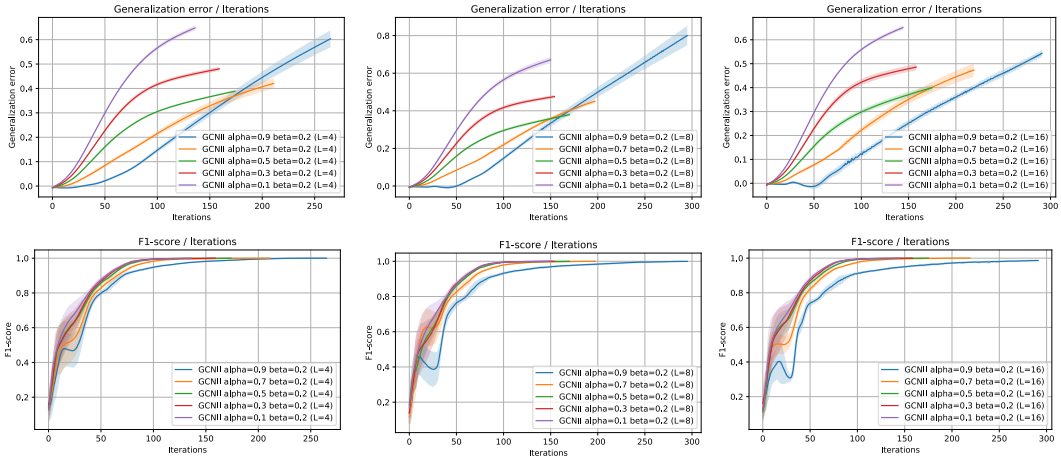

Figure 13: Comparison of α_ℓ on the generalization error on *Cora* dataset. The curve stops early at the largest training accuracy iteration.

E.3 Effect of DropEdge on generalization

In the following, we explore the effect of node embedding augmentation technique DropEdge [46] on the generalization of GCNs. Recall that DropEdge proposes to randomly drop a certain rate of edges in the input graph at each iteration and compute node embedding based on the sparser graph. The forward propagation rule of this technique can be formulated as $\mathbf{H}^{(\ell-1)} = \sigma(\widetilde{\mathbf{L}}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)}))$ where $\widetilde{\mathbf{L}}$ is constructed by the adjacency matrix of the sparser graph with $\text{supp}(\widetilde{\mathbf{L}}) \ll \text{supp}(\mathbf{L})$.

Again, we choose hidden dimension as 64, without applying dropout or weight decay during training. As shown in Figure 17 and Figure 18, DropEdge reduces the generalization error by restricting the number of nodes used during training. Besides, we observe that both training accuracy and generalization error decrease when the fraction of remaining edges in the graph decreases, which implies that edge dropping is impacting the generalization of GCN rather than the discriminativeness of node representations.

E.4 Effect of PairNorm on generalization

In the following, we explore the effect of node embedding augmentation technique PairNorm [60] on the generalization of GCNs. PairNorm proposes to normalized node embeddings by $\mathbf{H}^{(\ell)} =$

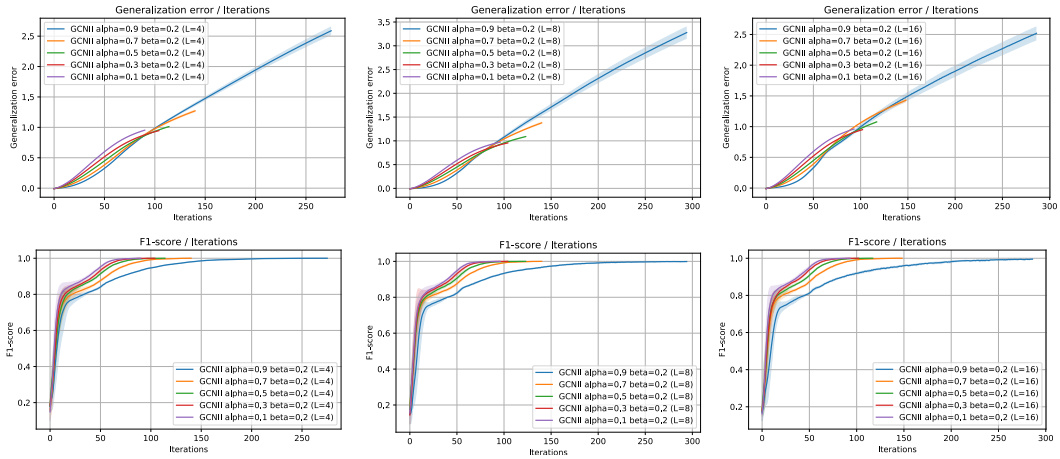

Figure 14: Comparison of α_ℓ on the generalization error on *Citeseer* dataset. The curve stops early at the largest training accuracy iteration.

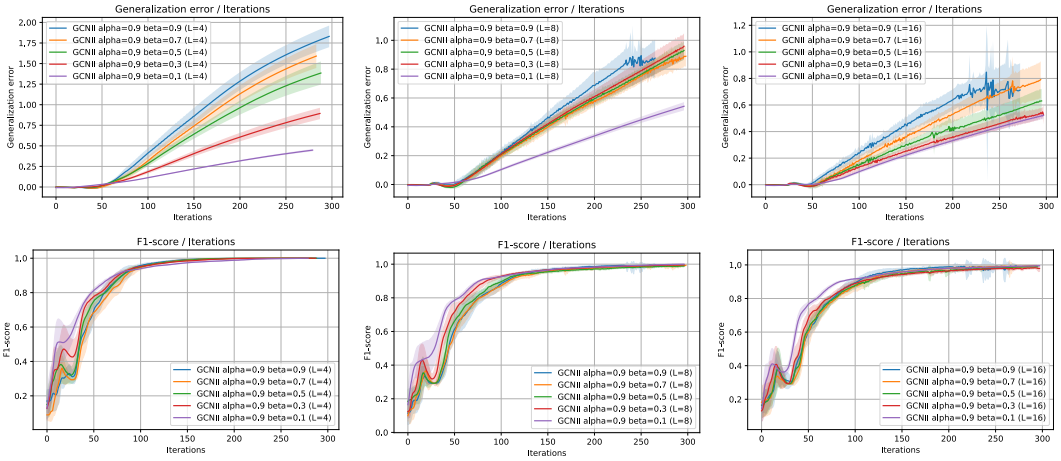

Figure 15: Comparison of β_ℓ on the generalization error on *Cora* dataset. The curve stops early at the largest training accuracy iteration.

$\texttt{PN}(\mathbf{H}^{(\ell)}) = \gamma \frac{\mathbf{H}^{(\ell)} - \boldsymbol{\mu}(\mathbf{H}^{(\ell)})}{\sigma(\mathbf{H}^{(\ell)})}$ where the average node embedding is computed as $\boldsymbol{\mu}(\mathbf{H}^{(\ell)}) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{h}_i^{(\ell)}$, the variance of node embeddings is computed as $\sigma^2(\mathbf{H}^{(\ell)}) = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{h}_i^{(\ell)} - \boldsymbol{\mu}(\mathbf{H}^{(\ell)})\|_2^2$, and $\gamma \geq 0$ controls the scale of node embeddings.

We choose hidden dimension as 64, and no dropout or weight decay is used during training. As shown in Figure 19 and Figure 20, a larger scale ratio γ can improve the discriminativeness of node embeddings, but will hurt the generalization error. A smaller scale ratio leads to a small generalization error, but it makes the node embeddings harder to discriminate, therefore over-smoothing happens (e.g., using $\gamma = 0.1$ can be think of as creating over-smoothing effect on node embeddings).

E.5 Illustrating the gradient instability issue during GCN training

Gradient instability refers to a phenomenon that the gradient changes significantly at every iteration. The main reason for gradient instability is because the scale of weight matrices is large, which causes the calculated gradients become large. Notice that the gradient instability issue is more significant on vanilla GCN than other sequential GCN structures such as ResGCN and GCNII, which is one of the key factors that impacts the training phase of vanilla GCNs.

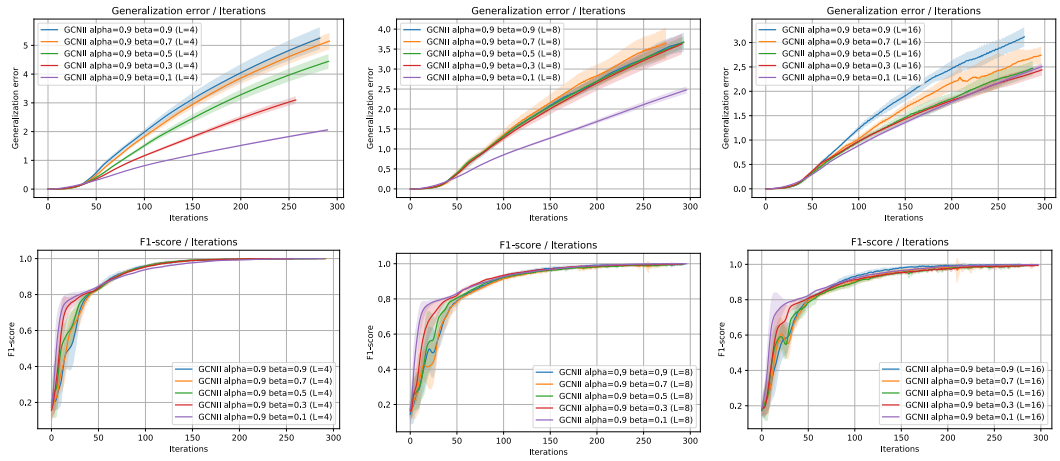

Figure 16: Comparison of β_ℓ on the generalization error on *Citeseer* dataset. The curve stops early at the largest training accuracy iteration.

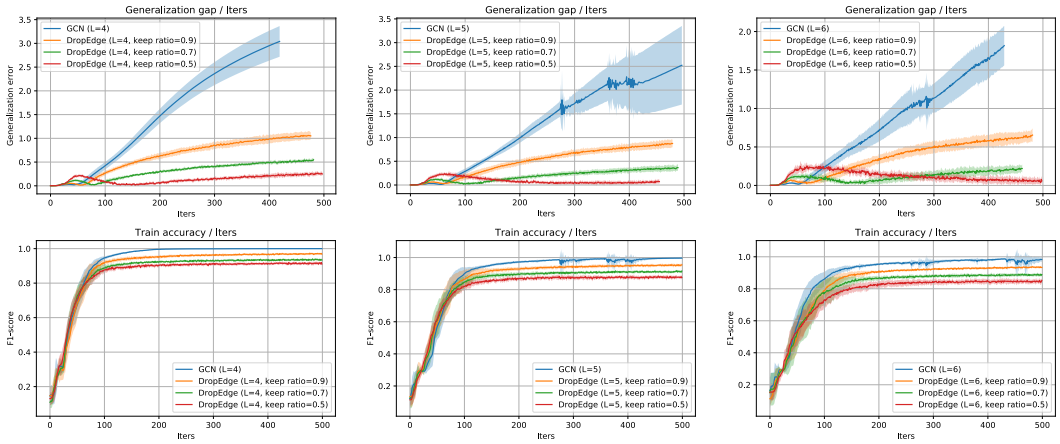

Figure 17: Comparison of generalization error of *DropEdge* on *Cora* dataset. The curve stops early at the largest training accuracy iteration.

To see this, let $\mathbf{W}^{(\ell)} \leftarrow \mathbf{W}^{(\ell)} - \eta \mathbf{G}^{(\ell)}$ denote the gradient descent update of the ℓth layer weight matrix, where $\mathbf{G}^{(\ell)}$ is the gradient with respect to the ℓth layer weight matrix $\mathbf{W}^{(\ell)}$. The upper bounds of the gradient for GCN, ResGCN, and GCNII are computed as

$$
\begin{aligned}
\|\mathbf{G}^{(\ell)}\|_2 &= \mathcal{O}\Big((\max\{1, \sqrt{d}B_w\})^L\Big), && \text{GCN (Eq. 71)} \\
\|\mathbf{G}^{(\ell)}\|_2 &= \mathcal{O}\Big((1 + \sqrt{d}B_w)^L\Big), && \text{ResGCN (Eq. 105)} \\
\|\mathbf{G}^{(\ell)}\|_2 &= \mathcal{O}\Big(\beta(\max\{1, \alpha\sqrt{d}B_w\})^L\Big), && \text{GCNII (Eq. 160)}
\end{aligned}
\tag{42}
$$

where d is the largest number of node degree, L is the number of layers, and $\|\mathbf{W}^{(\ell)}\|_2 \leq B_w$ is the largest singular value of weight matrix. Details please refer to the derivative of Eq. 71, Eq. 105 , and Eq. 160.

From Eq. 42, we know that the largest singular value of weight matrices is the key factor that affects the scale of the gradient. However, upper bound can be vacuous if we simply ignore the impact of network structure on B_w.

From Figure 21 and Figure 22, we can observe that the residual connection has implicit regularization on the weight matrices, which makes the weight matrices in ResGCN has a smaller largest singular values than GCN. As a result, the ResGCN does not suffer from gradient instability even its gradient

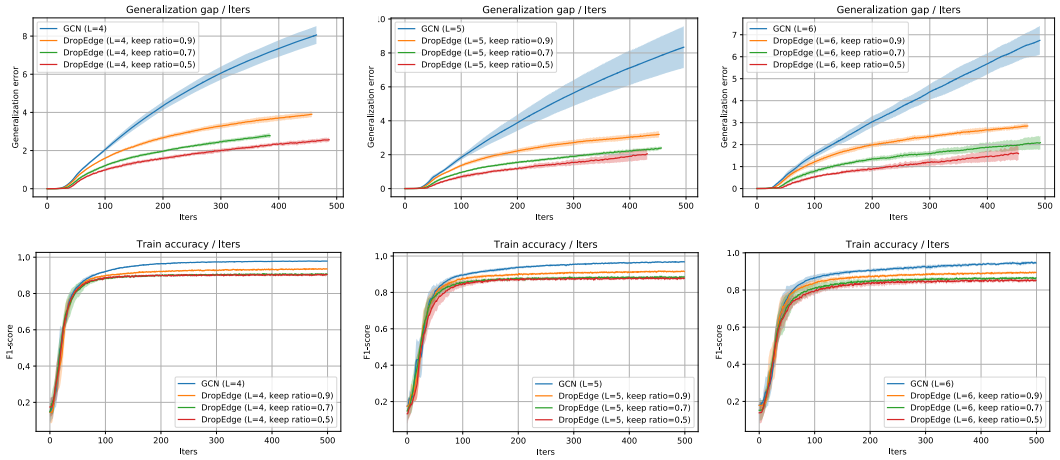

Figure 18: Comparison of generalization error of *DropEdge* on *Citeseer* dataset. The curve stops early at the largest training accuracy iteration.

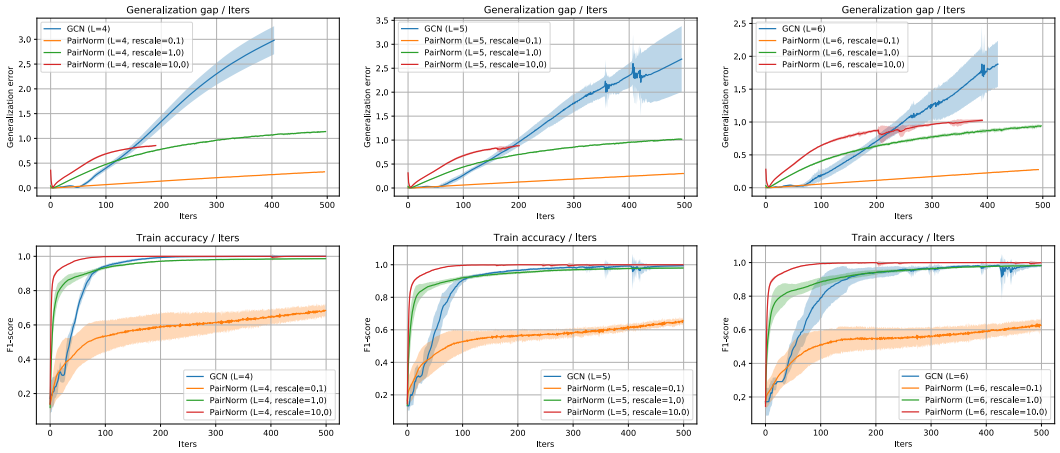

Figure 19: Comparison of generalization error of *PairNorm* on *Cora* dataset. The curve stops early at the largest training accuracy iteration.

norm upper bound in Eq. 42 is larger than GCN. Furthermore, although the largest singular value of the weight matrices for GCNII is larger than GCN, by selecting a small enough β_ℓ, GCNII can be less impacted by gradient instability than vanilla GCN.

E.6 Illustrating how more training leads to high training F1-score

As a compliment to Figure 1, we provide training and validation F1-score of the baseline models. During training, we chose hidden dimension as 64, Adam optimizer with learning rate 0.001, without any dropout or weight decay. Please note that removing dropout and weight decay is necessary because both operations are designed to prevent neural networks from overfitting, and will hurt the best training accuracy that a model can achieve. As shown in Figure 23 and Figure 24, all methods can achieve high training F1-score regardless the number of layers, which indicates node embeddings are distinguishable.

E.7 Effect of number of layers on real-word datasets

In the following, we demonstrate the effect of the number of layers and hyper-parameters on the performance of the model on OGB Arxiv [23]. We follow the default hyper-parameter setup of GCN

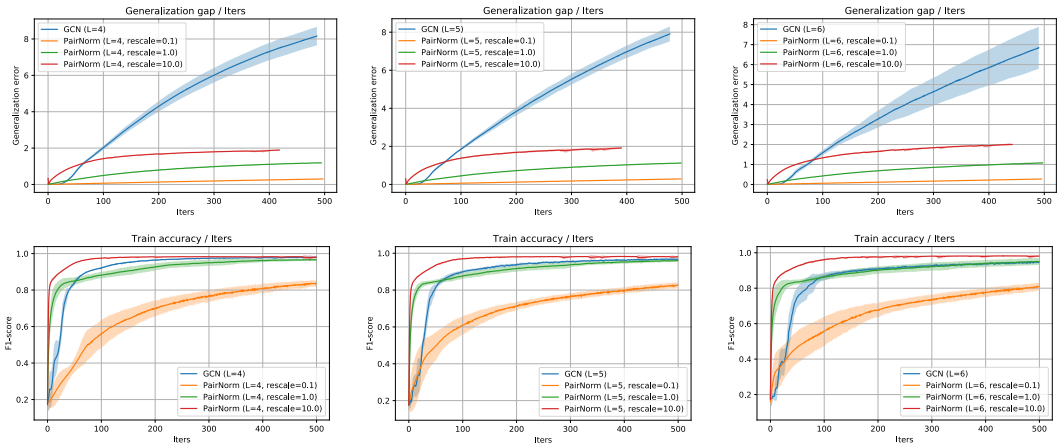

Figure 20: Comparison of generalization error of *PairNorm* on *Citeseer* dataset. The curve stops early at the largest training accuracy iteration.

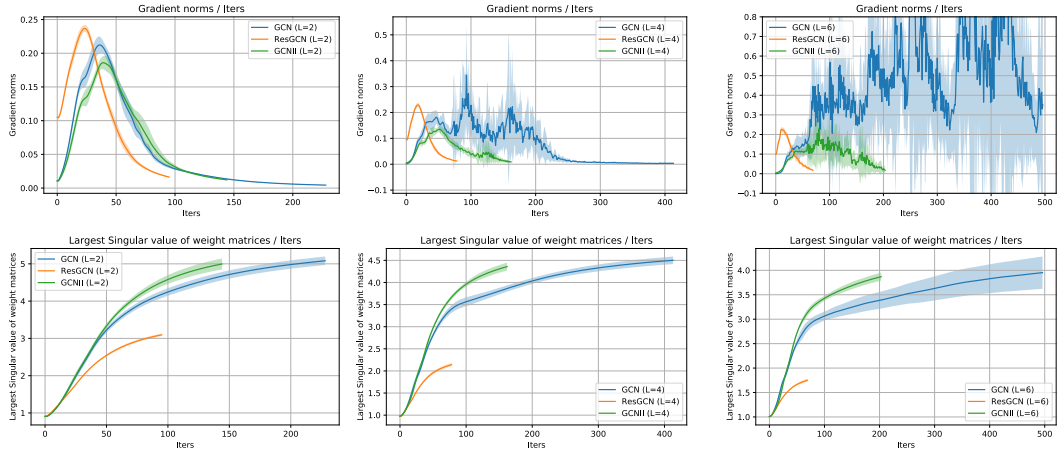

Figure 21: Comparison of gradient norm on *Cora* dataset. The curve stops early at the largest training accuracy iteration.

on the leaderboard,[10] i.e., we choose hidden dimension as 128, dropout ratio as 0.5, Adam optimizer with learning rate as 0.01, and applying batch normalization after each graph convolutional layer. As shown in Table 3, the number of layers and the choice of hyper-parameters can largely impact the performance of the models. Since DGCN can automatically adjust the α_ℓ and β_ℓ to better adapt to the change of model depth, it achieves a comparable and more stable performance than most baseline models.

F Generalization bound for GCN

In this section, we provide detailed proof on the generalization bound of GCN. Recall that the update rule of GCN is defined as

$$\mathbf{H}^{(\ell)} = \sigma(\mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)}), \tag{43}$$

where $\sigma(\cdot)$ is the ReLU activation function. Note that although ReLU function $\sigma(x)$ is not differentiable when $x = 0$, for analysis purpose we suppose the $\sigma'(0) = 0$.[11]

[10] https://github.com/snap-stanford/ogb/blob/master/examples/nodeproppred/arxiv/gnn.py. Due to the memory limitation, we choose hidden dimension as 128 instead of the default 256 hidden dimension for all models.

[11] Widely used deep learning frameworks, including PyTorch and Tensorflow, also set the subgradient of ReLU as zero when its input is zero.

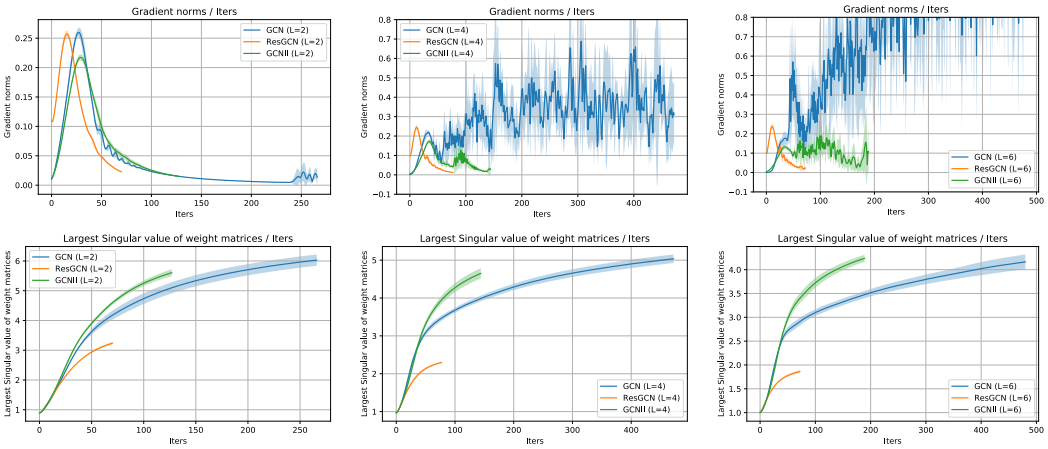

Figure 22: Comparison of gradient norm on *Citeseer* dataset. The curve stops early at the largest training accuracy iteration.

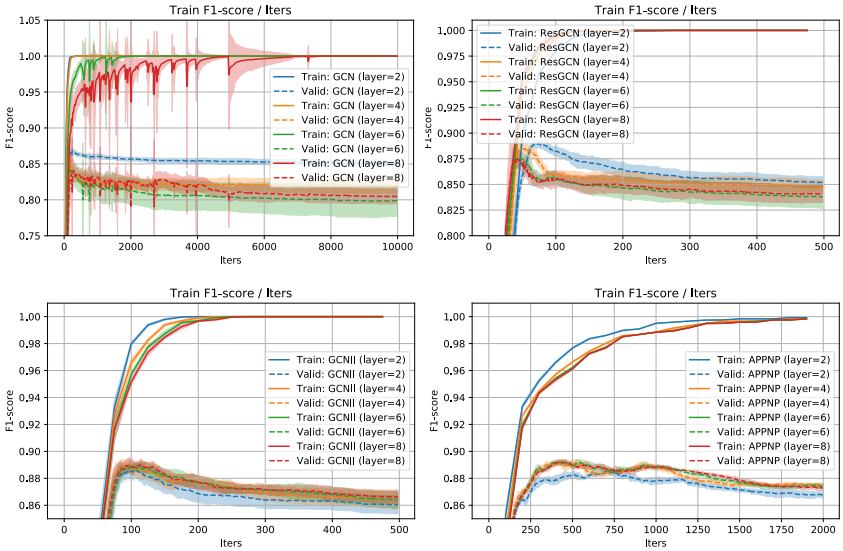

Figure 23: Comparison of training F1-score and number of iterations on *Cora* dataset.

The training of GCN is an empirical risk minimization with respect to a set of parameters $\boldsymbol{\theta} = \{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L)}, \mathbf{v}\}$, i.e.,

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m} \Phi_\gamma(-p(f(\mathbf{h}_i^{(L)}), y_i)),\ f(\mathbf{h}_i^{(L)}) = \tilde{\sigma}(\mathbf{v}^\top \mathbf{h}_i^{(L)}), \tag{44}$$

where $\mathbf{h}_i^{(L)}$ is the node representation of the ith node at the final layer, $f(\mathbf{h}_i^{(L)})$ is the predicted label for the ith node, $\tilde{\sigma}(x) = \frac{1}{\exp(-x)+1}$ is the sigmoid function, and loss function $\Phi_\gamma(-p(z, y))$ is $\frac{2}{\gamma}$-Lipschitz continuous with respect to its first input z with $p(z, y)$ as defined in Section 5. For simplification, we will use $\text{Loss}(z, y)$ which represents $\Phi_\gamma(-p(z, y))$ in the proof.

To establish the generalization of GCN as stated in Theorem 3, we utilize the following result on transductive uniform stability from [16].

Theorem 8 (Transductive uniform stability bound [16])**.** *Let f be a ϵ-uniformly stable transductive learner and $\gamma, \delta > 0$, and define $Q = mu/(m+u)$. Then, with probability at least $1-\delta$ over all*

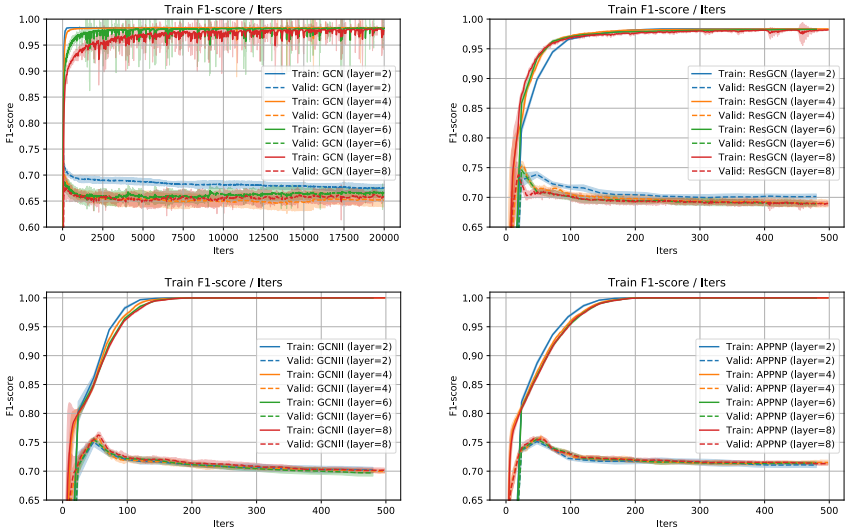

Figure 24: Comparison of training F1-score and number of iterations on *Citeseer* dataset.

Table 3: Comparison of F1-score on OGB-Arxiv dataset for different number of layers

Model	α	**2 Layers**	**4 Layers**	**8 Layers**	**12 Layers**	**16 Layers**
GCN	–	$71.02\% \pm 0.14$	$71.56\% \pm 0.19$	$71.28\% \pm 0.33$	$70.28\% \pm 0.23$	$69.37\% \pm 0.46$
ResGCN	–	$70.66\% \pm 0.48$	$72.41\% \pm 0.31$	$72.56\% \pm 0.31$	$72.46\% \pm 0.23$	$72.11\% \pm 0.28$
GCNII	0.9	$71.35\% \pm 0.21$	$72.57\% \pm 0.23$	$72.06\% \pm 0.42$	$71.31\% \pm 0.62$	$69.99\% \pm 0.80$
GCNII	0.8	$71.14\% \pm 0.27$	$72.32\% \pm 0.19$	$71.90\% \pm 0.41$	$71.21\% \pm 0.23$	$70.56\% \pm 0.72$
GCNII	0.5	$70.54\% \pm 0.30$	$72.09\% \pm 0.25$	$71.92\% \pm 0.32$	$71.24\% \pm 0.47$	$71.02\% \pm 0.58$
APPNP	0.9	$67.38\% \pm 0.34$	$68.02\% \pm 0.55$	$66.62\% \pm 0.48$	$67.43\% \pm 0.50$	$67.42\% \pm 1.00$
APPNP	0.8	$66.71\% \pm 0.32$	$68.25\% \pm 0.43$	$66.40\% \pm 0.89$	$66.51\% \pm 2.09$	$66.56\% \pm 0.74$
DGCN	–	$71.21\% \pm 0.25$	$72.29\% \pm 0.18$	$72.39\% \pm 0.21$	$\mathbf{72.63\% \pm 0.12}$	$72.41\% \pm 0.07$

training and testing partitions, we have

$$\mathcal{R}_u(f) \leq \mathcal{R}_m^{\gamma}(f) + \frac{2}{\gamma}\mathcal{O}\Big(\epsilon\sqrt{Q\ln(\delta^{-1})}\Big) + \mathcal{O}\Big(\frac{\ln(\delta^{-1})}{\sqrt{Q}}\Big).$$

Then, in Lemma 4, we derive the uniform stability constant for GCN, i.e., ϵ_{GCN}.

Lemma 4. *The uniform stability constant for GCN is computed as* $\epsilon_{\texttt{GCN}} = \frac{2\eta\rho_f G_f}{m}\sum_{t=1}^{T}(1+\eta L_F)^{t-1}$ *where*

$$\begin{aligned} &\rho_f = C_1^L C_2,\ G_f = \frac{2}{\gamma}(L+1)C_1^L C_2,\ L_f = \frac{2}{\gamma}(L+1)C_1^L C_2\Big((L+2)C_1^L C_2 + 2\Big), \\ &C_1 = \max\{1, \sqrt{d}B_w\},\ C_2 = \sqrt{d}(1+B_x). \end{aligned} \tag{45}$$

By plugging the result in Lemma 4 back to Theorem 8, we establish the generalization bound for GCN.

The key idea of the proof is to decompose the change of the GCN output into two terms (in Lemma 5) which depend on

- (Lemma 6) The maximum change of node embeddings, i.e., $\Delta h_{\max}^{(\ell)} = \max_i \|[\mathbf{H}^{(\ell)} - \tilde{\mathbf{H}}^{(\ell)}]_{i,:}\|_2$,
- (Lemma 7) The maximum node embeddings, i.e., $h_{\max}^{(\ell)} = \max_i \|[\mathbf{H}^{(\ell)}]_{i,:}\|_2$.

Lemma 5. *Let* $f(\mathbf{h}_i^{(L)}) = \tilde{\sigma}(\mathbf{v}^\top \mathbf{h}_i^{(L)})$, $\tilde{f}(\tilde{\mathbf{h}}_i^{(L)}) = \tilde{\sigma}(\tilde{\mathbf{v}}^\top \tilde{\mathbf{h}}_i^{(L)})$ *denote the prediction of node* i *using parameters* $\boldsymbol{\theta} = \{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L)}, \mathbf{v}\}, \tilde{\boldsymbol{\theta}} = \{\tilde{\mathbf{W}}^{(1)}, \ldots, \tilde{\mathbf{W}}^{(L)}, \tilde{\mathbf{v}}\}$ *(i.e., the two set of parameters*

trained on the original and the perturbed dataset) respectively. Then we have

$$\max_i |f(\mathbf{h}_i^{(L)}) - \tilde{f}(\tilde{\mathbf{h}}_i^{(L)})| \leq \Delta h_{\max}^{(L)} + h_{\max}^{(L)} \|\Delta \mathbf{v}\|_2,$$
$$\max_i \left\| \frac{\partial f(\mathbf{h}_i^{(L)})}{\partial \mathbf{h}_i^{(L)}} - \frac{\partial \tilde{f}(\tilde{\mathbf{h}}_i^{(L)})}{\partial \tilde{\mathbf{h}}_i^{(L)}} \right\|_2 \leq \Delta h_{\max}^{(L)} + (h_{\max}^{(L)} + 1)\|\Delta \mathbf{v}\|_2, \tag{46}$$

where $\Delta \mathbf{v} = \mathbf{v} - \tilde{\mathbf{v}}$.

Lemma 6 (Upper bound of $h_{\max}^{(\ell)}$ for **GCN**)**.** *Let suppose Assumption 1 hold. Then, the maximum node embeddings for any node at the* ℓ*th layer is bounded by*

$$h_{\max}^{(\ell)} \leq B_x(\max\{1, \sqrt{d}B_w\})^{\ell}. \tag{47}$$

Lemma 7 (Upper bound of $\Delta h_{\max}^{(\ell)}$ for **GCN**)**.** *Let suppose Assumption 1 hold. Then, the maximum change between the node embeddings on two different set of weight parameters for any node at the* ℓ*th layer is bounded by*

$$\Delta h_{\max}^{(\ell)} \leq \sqrt{d}B_x(\max\{1, \sqrt{d}B_w\})^{\ell-1}(\|\Delta \mathbf{W}^{(1)}\|_2 + \ldots + \|\Delta \mathbf{W}^{(\ell)}\|_2), \tag{48}$$

where $\Delta \mathbf{W}^{(\ell)} = \mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)}$.

Besides, in Lemma 8, we derive the upper bound on the maximum change of node embeddings before the activation function, i.e., $\Delta z_{\max}^{(\ell)} = \max_i \|[\mathbf{Z}^{(\ell)} - \tilde{\mathbf{Z}}^{(\ell)}]_{i,:}\|_2$. $\Delta z_{\max}^{(\ell)}$ will be used in the computation of the gradient related upper bounds.

Lemma 8 (Upper bound of $\Delta z_{\max}^{(\ell)}$ for **GCN**)**.** *Let suppose Assumption 1 hold. Then, the maximum change between the node embeddings before the activation function on two different set of weight parameters for any node at the* ℓ*th layer is bounded by*

$$\Delta z_{\max}^{(\ell)} \leq \sqrt{d}B_x(\max\{1, \sqrt{d}B_w\})^{\ell-1}\Big(\|\Delta \mathbf{W}^{(1)}\|_2 + \ldots + \|\Delta \mathbf{W}^{(\ell)}\|_2\Big), \tag{49}$$

where $\Delta \mathbf{W}^{(\ell)} = \mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)}$.

Then, in Lemma 9, we decompose the change of the model parameters into two terms which depend on

- The maximum change of gradient passing from the $(\ell+1)$th layer to the ℓth layer $\Delta d_{\max}^{(\ell)} = \max_i \left\| \left[\frac{\partial \sigma(\mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)})}{\partial \mathbf{H}^{(\ell-1)}} - \frac{\partial \sigma(\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}\tilde{\mathbf{W}}^{(\ell)})}{\partial \tilde{\mathbf{H}}^{(\ell-1)}} \right]_{i,:} \right\|_2$,
- The maximum gradient passing from the $(\ell+1)$th layer to the ℓth layer $d_{\max}^{(\ell)} = \max_i \left\| \left[\frac{\partial \sigma(\mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)})}{\partial \mathbf{H}^{(\ell-1)}} \right]_{i,:} \right\|_2$.

Lemma 9 (Upper bound of $d_{\max}^{(\ell)}, \Delta d_{\max}^{(\ell)}$ for **GCN**)**.** *Let suppose Assumption 1 hold. Then, the upper bound on the maximum gradient passing from layer* $\ell+1$ *to layer* ℓ *for any node is*

$$d_{\max}^{(\ell)} \leq \frac{2}{\gamma}(\max\{1, \sqrt{d}B_w\})^{L-\ell+1}, \tag{50}$$

and the upper bound on the maximum change between the gradient passing from layer $\ell+1$ *to layer* ℓ *on two different set of weight parameters for any node is*

$$\Delta d_{\max}^{(\ell)} \leq (\max\{1, \sqrt{d}B_w\})^{L-\ell}\frac{2}{\gamma}\Big((L+1)\sqrt{d}(B_x+1)(\max\{1, \sqrt{d}B_w\})^L + 1\Big)\|\Delta \boldsymbol{\theta}\|_2, \tag{51}$$

where $\|\Delta \boldsymbol{\theta}\|_2 = \|\mathbf{v} - \tilde{\mathbf{v}}\|_2 + \sum_{\ell-1}^{L} \|\mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)}\|_2$ *denotes the change of two set of parameters.*

Finally, based on the previous result, in Lemma 10, we decompose the change of the model parameters into two terms which depend on

- The change of gradient with respect to the ℓth layer weight parameters $\|\Delta \mathbf{G}^{(\ell)}\|_2$,

- The gradient with respect to the ℓth layer weight parameters $\|\mathbf{G}^{(\ell)}\|_2$,

where $\mathbf{G}^{(L+1)}$ denotes the gradient with respect to the weight $\mathbf{v}$ of the binary classifier and $\mathbf{G}^{(\ell)}$ denotes the gradient with respect to the weight $\mathbf{W}^{(\ell)}$ of the ℓth graph convolutional layer. Notice that $\|\Delta\mathbf{G}^{(\ell)}\|_2$ reflects the smoothness of GCN model and $\|\mathbf{G}^{(\ell)}\|_2$ corresponds to the upper bound of gradient.

Lemma 10 (Upper bound of $\|\mathbf{G}^{(\ell)}\|_2, \|\Delta\mathbf{G}^{(\ell)}\|_2$ for **GCN**). *Let suppose Assumption 1 hold and let $C_1 = \max\{1, \sqrt{d}B_w\}$ and $C_2 = \sqrt{d}(B_x + 1)$. Then, the gradient and the maximum change between gradients computed on two different set of weight parameters are bounded by*

$$\begin{aligned}
&\sum_{\ell=1}^{L+1} \|\mathbf{G}^{(\ell)}\|_2 \leq \frac{2}{\gamma}(L+1)C_1^L C_2 \|\Delta\boldsymbol{\theta}\|_2, \\
&\sum_{\ell=1}^{L+1} \|\Delta\mathbf{G}^{(\ell)}\|_2 \leq \frac{2}{\gamma}(L+1)C_1^L C_2 \Big((L+2)C_1^L C_2 + 2\Big)\|\Delta\boldsymbol{\theta}\|_2,
\end{aligned} \tag{52}$$

where $\|\Delta\boldsymbol{\theta}\|_2 = \|\mathbf{v} - \tilde{\mathbf{v}}\|_2 + \sum_{\ell-1}^{L} \|\mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)}\|_2$ denotes the change of two set of parameters.

Equipped with above intermediate results, we now proceed to prove Lemma 4.

Proof of Lemma 4. Recall that our goal is to explore the impact of different GCN structures on the uniform stability constant ϵ_{GCN}, which is a function of ρ_f, G_f, and L_f. Let $C_1 = \max\{1, \sqrt{d}B_w\}$ and $C_2 = \sqrt{d}(B_x + 1)$. Firstly, by plugging Lemma 7 and Lemma 6 into Lemma 5, we have

$$\begin{aligned}
\max_i |f(\mathbf{h}_i^{(L)}) - \tilde{f}(\tilde{\mathbf{h}}_i^{(L)})| &\leq \Delta h_{\max}^{(L)} + h_{\max}^{(L)} \|\Delta\mathbf{v}\|_2 \\
&\leq \sqrt{d}B_x \cdot (\max\{1, \sqrt{d}B_w\})^L \|\Delta\boldsymbol{\theta}\|_2 \\
&\leq C_1^L C_2 \|\Delta\boldsymbol{\theta}\|_2.
\end{aligned} \tag{53}$$

Therefore, we know that the function f is ρ_f-Lipschitz continuous, with $\rho_f = C_1^L C_2$. Then, by Lemma 10, we know that the function f is L_f-smoothness, and the gradient of each weight matrix is bounded by G_f, with

$$G_f = \frac{2}{\gamma}(L+1)C_1^L C_2, \; L_f = \frac{2}{\gamma}(L+1)C_1^L C_2 \Big((L+2)C_1^L C_2 + 2\Big). \tag{54}$$

By plugging ϵ_{GCN} into Theorem 3, we obtain the generalization bound of GCN. □

F.1 Proof of Lemma 5

By the definition of $f(\mathbf{h}_i^{(L)})$ and $\tilde{f}(\tilde{\mathbf{h}}_i^{(L)})$, we have

$$\begin{aligned}
\max_i |f(\mathbf{h}_i^{(L)}) - \tilde{f}(\tilde{\mathbf{h}}_i^{(L)})| &= \max_i |\tilde{\sigma}(\mathbf{v}^\top \mathbf{h}_i^{(L)}) - \tilde{\sigma}(\tilde{\mathbf{v}}^\top \tilde{\mathbf{h}}_i^{(L)})| \\
&\underset{(a)}{\leq} \max_i \|\mathbf{v}^\top \mathbf{h}_i^{(L)} - \tilde{\mathbf{v}}^\top \tilde{\mathbf{h}}_i^{(L)}\|_2 \\
&\leq \max_i \|\mathbf{v}^\top (\mathbf{h}_i^{(L)} - \tilde{\mathbf{h}}_i^{(L)})\|_2 + \max_i \|\tilde{\mathbf{h}}_i^{(L)}(\mathbf{v} - \tilde{\mathbf{v}})\|_2 \\
&\leq \Delta h_{\max}^{(L)} + h_{\max}^{(L)} \Delta\mathbf{v},
\end{aligned} \tag{55}$$

where (a) is due to the fact that sigmoid function is 1-Lipschitz continuous.

$$\begin{aligned}
\max_i \left\| \frac{\partial f(\mathbf{h}_i^{(L)})}{\partial \mathbf{h}_i^{(L)}} - \frac{\partial \tilde{f}(\tilde{\mathbf{h}}_i^{(L)})}{\partial \tilde{\mathbf{h}}_i^{(L)}} \right\|_2 &= \max_i \|\tilde{\sigma}'(\mathbf{v}^\top \mathbf{h}_i^{(L)})\mathbf{v}^\top - \tilde{\sigma}'(\tilde{\mathbf{v}}^\top \tilde{\mathbf{h}}_i^{(L)})\tilde{\mathbf{v}}^\top\|_2 \\
&\leq \max_i \|\tilde{\sigma}'(\mathbf{v}^\top \mathbf{h}_i^{(L)})\mathbf{v}^\top - \tilde{\sigma}'(\mathbf{v}^\top \mathbf{h}_i^{(L)})\tilde{\mathbf{v}}^\top\|_2 \\
&\quad + \max_i \|\tilde{\sigma}'(\mathbf{v}^\top \mathbf{h}_i^{(L)})\tilde{\mathbf{v}}^\top - \tilde{\sigma}'(\tilde{\mathbf{v}}^\top \tilde{\mathbf{h}}_i^{(L)})\tilde{\mathbf{v}}^\top\|_2 \\
&\underset{(a)}{\leq} \Delta \mathbf{v} + \left(\Delta h_{\max}^{(L)} + h_{\max}^{(L)} \Delta \mathbf{v}\right) \\
&= \Delta h_{\max}^{(L)} + (h_{\max}^{(L)} + 1)\Delta \mathbf{v},
\end{aligned} \tag{56}$$

where (a) is due to the fact that sigmoid function and its gradient are 1-Lipschitz continuous.

F.2 Proof of Lemma 6

By the definition of $h_{\max}^{(\ell)}$, we have

$$\begin{aligned}
h_{\max}^{(\ell)} &= \max_i \|[\sigma(\mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)})]_{i,:}\|_2 \\
&\underset{(a)}{\leq} \max_i \|[\mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)}]_{i,:}\|_2 \\
&\leq \max_i \|[\mathbf{L}\mathbf{H}^{(\ell-1)}]_{i,:}\|_2 \|\mathbf{W}^{(\ell)}\|_2 \\
&= \max_i \left\| \sum_{j=1}^N L_{i,j} \mathbf{h}_j^{(\ell-1)} \right\|_2 \|\mathbf{W}^{(\ell)}\|_2 \\
&\leq \max_i \| \sum_{j=1}^N L_{i,j} \|_2 \cdot \max_j \|\mathbf{h}_j^{(\ell-1)}\|_2 \cdot \|\mathbf{W}^{(\ell)}\|_2 \\
&\underset{(b)}{\leq} \sqrt{d} \|\mathbf{W}^{(\ell)}\|_2 \cdot h_{\max}^{(\ell-1)} \\
&\leq \sqrt{d} B_w \cdot h_{\max}^{(\ell-1)} \leq (\max\{1, \sqrt{d} B_w))^\ell B_x,
\end{aligned} \tag{57}$$

where (a) is due to $\|\sigma(\mathbf{x})\|_2 \leq \|\mathbf{x}\|_2$ and (b) is due to Lemma 31.

F.3 Proof of Lemma 7

By the definition of $\Delta h_{\max}^{(\ell)}$, we have

$$\begin{aligned}
\Delta h_{\max}^{(\ell)} &= \max_i \|\mathbf{h}_i^{(\ell)} - \tilde{\mathbf{h}}_i^{(\ell)}\|_2 \\
&= \max_i \|[\sigma(\mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)}) - \sigma(\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}\tilde{\mathbf{W}}^{(\ell)})]_{i,:}\|_2 \\
&= \max_i \|[\mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)} - \mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}\tilde{\mathbf{W}}^{(\ell)}]_{i,:}\|_2 \\
&\leq \max_i \|[\mathbf{L}\mathbf{H}^{(\ell-1)}(\mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)}) + \mathbf{L}(\mathbf{H}^{(\ell-1)} - \tilde{\mathbf{H}}^{(\ell-1)})\tilde{\mathbf{W}}^{(\ell)}]_{i,:}\|_2 \\
&\leq \max_i \left\| \sum_{j=1}^N L_{i,j} \mathbf{h}_j^{(\ell-1)} \right\|_2 \|\Delta \mathbf{W}^{(\ell)}\|_2 + \max_i \left\| \sum_{j=1}^N L_{i,j} \Delta \mathbf{h}_j^{(\ell-1)} \right\|_2 \|\tilde{\mathbf{W}}^{(\ell)}\|_2 \\
&\leq \sqrt{d} \Delta h_{\max}^{(\ell-1)} \|\tilde{\mathbf{W}}^{(\ell)}\|_2 + \sqrt{d} h_{\max}^{(\ell-1)} \|\Delta \mathbf{W}^{(\ell)}\|_2 \\
&\leq \sqrt{d} B_w \Delta h_{\max}^{(\ell-1)} + \sqrt{d} h_{\max}^{(\ell-1)} \|\Delta \mathbf{W}^{(\ell)}\|_2.
\end{aligned} \tag{58}$$

By induction, we have

$$
\begin{aligned}
\Delta h_{\max}^{(\ell)} &\leq \sqrt{d}B_w \Delta h_{\max}^{(\ell-1)} + \sqrt{d} h_{\max}^{(\ell-1)} \|\Delta \mathbf{W}^{(\ell)}\|_2 \\
&\leq (\sqrt{d}B_w)^2 \Delta h_{\max}^{(\ell-2)} + \sqrt{d}\Big(h_{\max}^{(\ell-1)} \|\Delta \mathbf{W}^{(\ell)}\|_2 + (\sqrt{d}B_w) h_{\max}^{(\ell-2)} \|\Delta \mathbf{W}^{(\ell-1)}\|_2 \Big) \\
&\dots \\
&\leq (\sqrt{d}B_w)^\ell \Delta h_{\max}^{(0)} + \sqrt{d}\Big(h_{\max}^{(\ell-1)} \|\Delta \mathbf{W}^{(\ell)}\|_2 + (\sqrt{d}B_w) h_{\max}^{(\ell-2)} \|\Delta \mathbf{W}^{(\ell-1)}\|_2 + \ldots + (\sqrt{d}B_w)^{\ell-1} h_{\max}^{(0)} \|\Delta \mathbf{W}^{(1)}\|_2 \Big) \\
&\underset{(a)}{=} \sqrt{d}\Big(h_{\max}^{(\ell-1)} \|\Delta \mathbf{W}^{(\ell)}\|_2 + (\sqrt{d}B_w) h_{\max}^{(\ell-2)} \|\Delta \mathbf{W}^{(\ell-1)}\|_2 + \ldots + (\sqrt{d}B_w)^{\ell-1} h_{\max}^{(0)} \|\Delta \mathbf{W}^{(1)}\|_2 \Big),
\end{aligned}
\tag{59}
$$

where (a) is due to $\Delta h_{\max}^{(0)} = 0$. Plugging in the upper bound of $h_{\max}^{(\ell)}$ in Lemma 6, yields

$$
\begin{aligned}
\Delta h_{\max}^{(\ell)} &\leq \sqrt{d}B_x (\sqrt{d}B_w)^{\ell-1} \Big(\|\Delta \mathbf{W}^{(\ell)}\|_2 + \ldots + \|\Delta \mathbf{W}^{(1)}\|_2) \\
&\leq \sqrt{d}B_x (\max\{1, \sqrt{d}B_w\})^{\ell-1} \Big(\|\Delta \mathbf{W}^{(\ell)}\|_2 + \ldots + \|\Delta \mathbf{W}^{(1)}\|_2).
\end{aligned}
\tag{60}
$$

F.4 Proof of Lemma 8

By the definition of $\mathbf{Z}^{(\ell)}$, we have

$$
\begin{aligned}
\mathbf{Z}^{(\ell)} - \tilde{\mathbf{Z}}^{(\ell)} &= \mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)} - \mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}\tilde{\mathbf{W}}^{(\ell)} \\
&= \mathbf{L}\Big(\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)} - \tilde{\mathbf{H}}^{(\ell-1)}\mathbf{W}^{(\ell)} + \tilde{\mathbf{H}}^{(\ell-1)}\mathbf{W}^{(\ell)} - \tilde{\mathbf{H}}^{(\ell-1)}\tilde{\mathbf{W}}^{(\ell)} \Big) \\
&= \mathbf{L}\big(\mathbf{H}^{(\ell-1)} - \tilde{\mathbf{H}}^{(\ell-1)} \big)\mathbf{W}^{(\ell)} + \mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}\big(\mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)} \big).
\end{aligned}
\tag{61}
$$

By taking norm on the both side of the equation, we have

$$
\begin{aligned}
\Delta z_{\max}^{(\ell)} &= \max_i \|[\mathbf{Z}^{(\ell)} - \tilde{\mathbf{Z}}^{(\ell)}]_{i,:}\|_2 \\
&\leq \sqrt{d}B_w \cdot \max_i \|[\mathbf{Z}^{(\ell-1)} - \tilde{\mathbf{Z}}^{(\ell-1)}]_{i,:}\|_2 + \sqrt{d} h_{\max}^{(\ell-1)} \|\Delta \mathbf{W}^{(\ell)}\|_2 \\
&= \sqrt{d}B_w \cdot z_{\max}^{(\ell-1)} + \sqrt{d} h_{\max}^{(\ell-1)} \|\Delta \mathbf{W}^{(\ell)}\|_2.
\end{aligned}
\tag{62}
$$

By induction, we have

$$
\begin{aligned}
\Delta z_{\max}^{(\ell)} &\leq \sqrt{d}B_w \cdot \Delta z_{\max}^{(\ell-1)} + \sqrt{d} h_{\max}^{(\ell-1)} \|\Delta \mathbf{W}^{(\ell)}\|_2 \\
&\underset{(a)}{\leq} \sqrt{d}B_w \cdot \Delta z_{\max}^{(\ell-1)} + \sqrt{d}B_x (\sqrt{d}B_w)^{\ell-1} \|\Delta \mathbf{W}^{(\ell)}\|_2 \\
&\leq \sqrt{d}B_w \Big(\sqrt{d}B_w \cdot \Delta z_{\max}^{(\ell-2)} + \sqrt{d}B_x (\sqrt{d}B_w)^{\ell-2} \|\Delta \mathbf{W}^{(\ell-1)}\|_2 \Big) + \sqrt{d}B_x (\sqrt{d}B_w)^{\ell-1} \|\Delta \mathbf{W}^{(\ell)}\|_2 \\
&= (\sqrt{d}B_w)^2 \cdot \Delta z_{\max}^{(\ell-2)} + \sqrt{d}B_x (\sqrt{d}B_w)^{\ell-1} \Big(\|\Delta \mathbf{W}^{(\ell-1)}\|_2 + \|\Delta \mathbf{W}^{(\ell)}\|_2 \Big) \\
&\leq \dots \\
&\leq \sqrt{d}B_x (\sqrt{d}B_w)^{\ell-1} \Big(\|\Delta \mathbf{W}^{(1)}\|_2 + \ldots + \|\Delta \mathbf{W}^{(\ell)}\|_2 \Big),
\end{aligned}
\tag{63}
$$

where (a) is due to $h_{\max}^{(\ell-1)} \leq (\sqrt{d}B_w)^{\ell-1} B_x$.

F.5 Proof of Lemma 9

For notation simplicity, let $\mathbf{D}^{(\ell)}$ denote the gradient passing from ℓth to $(\ell-1)$th layer. By the definition of $d_{\max}^{(\ell)}$, we have

$$\begin{aligned}
d_{\max}^{(\ell)} &= \max_i \|\mathbf{d}_i^{(\ell)}\|_2 \\
&= \max_i \|[\mathbf{L}^\top \sigma'(\mathbf{Z}^{(\ell)}) \odot \mathbf{D}^{(\ell+1)}\mathbf{W}^{(\ell)}]_{i,:}\|_2 \\
&\underset{(a)}{\leq} \max_i \|[\mathbf{L}^\top \mathbf{D}^{(\ell+1)}]_{i,:}\|_2 \|\mathbf{W}^{(\ell)}\|_2 \\
&\leq \max_i \left\| \sum_{j=1}^N L_{i,j}^\top \mathbf{d}_j^{(\ell+1)} \right\|_2 \|\mathbf{W}^{(\ell)}\|_2 \\
&\leq \sqrt{d}\|\mathbf{W}^{(\ell)}\|_2 \cdot d_{\max}^{(\ell+1)} \\
&\leq \sqrt{d}B_w \cdot d_{\max}^{(\ell+1)} \\
&\underset{(b)}{\leq} \frac{2}{\gamma}(\sqrt{d}B_w)^{L-\ell+1} \leq \frac{2}{\gamma}(\max\{1, \sqrt{d}B_w\})^{L-\ell+1}
\end{aligned} \tag{64}$$

where (a) is due to each element in $\sigma'(\mathbf{Z}^{(\ell)})$ is either 0 or 1 depending on $\mathbf{Z}^{(\ell)}$, and (b) is due to $\|\frac{\partial \text{Loss}(f(\mathbf{h}_i^{(L)}), y_i)}{\partial \mathbf{h}_i^{(L)}}\|_2 \leq 2/\gamma$. By taking norm on the both sides, we have

$$\begin{aligned}
\Delta d_{\max}^{(\ell)} &= \max_i \|[\mathbf{D}^{(\ell)} - \tilde{\mathbf{D}}^{(\ell)}]_{i,:}\|_2 \\
&= \max_i \|[\mathbf{L}^\top(\mathbf{D}^{(\ell+1)} \odot \sigma'(\mathbf{Z}^{(\ell)}))[\mathbf{W}^{(\ell)}]^\top - \mathbf{L}^\top(\tilde{\mathbf{D}}^{(\ell+1)} \odot \sigma'(\tilde{\mathbf{Z}}^{(\ell)}))[\tilde{\mathbf{W}}^{(\ell)}]^\top]_{i,:}\|_2 \\
&\leq \max_i \|[\mathbf{L}^\top(\mathbf{D}^{(\ell+1)} \odot \sigma'(\mathbf{Z}^{(\ell)}))[\mathbf{W}^{(\ell)}]^\top - \mathbf{L}^\top(\tilde{\mathbf{D}}^{(\ell+1)} \odot \sigma'(\mathbf{Z}^{(\ell)}))[\mathbf{W}^{(\ell)}]^\top]_{i,:}\|_2 \\
&\quad + \max_i \|[\mathbf{L}^\top(\tilde{\mathbf{D}}^{(\ell+1)} \odot \sigma'(\mathbf{Z}^{(\ell)}))[\mathbf{W}^{(\ell)}]^\top - \mathbf{L}^\top(\tilde{\mathbf{D}}^{(\ell+1)} \odot \sigma'(\tilde{\mathbf{Z}}^{(\ell)}))[\mathbf{W}^{(\ell)}]^\top]_{i,:}\|_2 \\
&\quad + \max_i \|[\mathbf{L}^\top(\tilde{\mathbf{D}}^{(\ell+1)} \odot \sigma'(\tilde{\mathbf{Z}}^{(\ell)}))[\mathbf{W}^{(\ell)}]^\top - \mathbf{L}^\top(\tilde{\mathbf{D}}^{(\ell+1)} \odot \sigma'(\tilde{\mathbf{Z}}^{(\ell)}))[\tilde{\mathbf{W}}^{(\ell)}]^\top]_{i,:} \\
&\underset{(a)}{\leq} \max_i \|[\mathbf{L}^\top((\mathbf{D}^{(\ell+1)} - \tilde{\mathbf{D}}^{(\ell+1)}))[\mathbf{W}^{(\ell)}]^\top]_{i,:}\|_2 \\
&\quad + \max_i \|[\mathbf{L}^\top(\tilde{\mathbf{D}}^{(\ell+1)}[\mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)}]^\top]_{i,:}\|_2 \\
&\quad + \max_i \|[\mathbf{L}\tilde{\mathbf{D}}^{(\ell+1)}(\mathbf{Z}^{(\ell)} - \tilde{\mathbf{Z}}^{(\ell)})\mathbf{W}^{(\ell)}]_{i,:}\|_2 \\
&\leq \sqrt{d}B_w \Delta d_{\max}^{(\ell+1)} + \sqrt{d} d_{\max}^{(\ell+1)} \|\Delta \mathbf{W}^{(\ell)}\|_2 + \sqrt{d}B_w d_{\max}^{(\ell+1)} \Delta z_{\max}^{(\ell)} \\
&\leq \sqrt{d}B_w \Delta d_{\max}^{(\ell+1)} + \underbrace{d_{\max}^{(\ell+1)}\Big(\sqrt{d}\|\Delta \mathbf{W}^{(\ell)}\|_2 + \sqrt{d}B_w \Delta z_{\max}^{(\ell)}\Big)}_{(A)},
\end{aligned} \tag{65}$$

where inequality (a) is due to the gradient of ReLU activation function is either 1 or 0.

Knowing that $d_{\max}^{(\ell+1)} \leq \frac{2}{\gamma}(\sqrt{d}B_w)^{L-\ell}$ and using the upper bound of $z_{\max}^{(\ell)}$ as derived in Lemma 8, we can upper bound (A) as

$$\begin{aligned}
&d_{\max}^{(\ell+1)}\Big(\sqrt{d}\|\Delta \mathbf{W}^{(\ell)}\|_2 + \sqrt{d}B_w \Delta z_{\max}^{(\ell)}\Big) \\
&\quad \leq \frac{2}{\gamma}(\sqrt{d}B_w)^{L-\ell}\Big(\sqrt{d}\|\Delta \mathbf{W}^{(\ell)}\|_2 + \sqrt{d}B_x(\sqrt{d}B_w)^\ell\big(\|\Delta \mathbf{W}^{(1)}\|_2 + \ldots + \|\Delta \mathbf{W}^{(\ell)}\|_2\big)\Big) \\
&\quad \leq \frac{2}{\gamma}(\sqrt{d}B_w)^{L-\ell}\Big(\sqrt{d} + \sqrt{d}B_x(\sqrt{d}B_w)^\ell\Big)\big(\|\Delta \mathbf{W}^{(1)}\|_2 + \ldots + \|\Delta \mathbf{W}^{(\ell)}\|_2\big) \\
&\quad \leq \frac{2}{\gamma}(\sqrt{d}B_w)^{L-\ell}\sqrt{d}(1+B_x)(\max\{1, \sqrt{d}B_w\})^\ell \|\Delta \boldsymbol{\theta}\|_2 \\
&\quad \leq \frac{2}{\gamma}\sqrt{d}(1+B_x)(\max\{1, \sqrt{d}B_w\})^L \|\Delta \boldsymbol{\theta}\|_2.
\end{aligned} \tag{66}$$

Plugging it back, we have

$$
\begin{aligned}
\Delta d_{\max}^{(\ell)} &\leq \sqrt{d}B_w \Delta d_{\max}^{(\ell+1)} + \frac{2}{\gamma}\sqrt{d}(B_x+1)(\max\{1,\sqrt{d}B_w\})^L \|\Delta\boldsymbol{\theta}\|_2 \\
&\leq \sqrt{d}B_w\Big(\sqrt{d}B_w \Delta d_{\max}^{(\ell+2)} + \frac{2}{\gamma}\sqrt{d}(B_x+1)(\max\{1,\sqrt{d}B_w\})^L \|\Delta\boldsymbol{\theta}\|_2\Big) \\
&\quad + \frac{2}{\gamma}\sqrt{d}(B_x+1)(\max\{1,\sqrt{d}B_w\})^L \|\Delta\boldsymbol{\theta}\|_2 \\
&\leq \Big(1+\sqrt{d}B_w + \ldots + (\max\{1,\sqrt{d}B_w\})^{L-\ell-1}\Big)\cdot \frac{2}{\gamma}\sqrt{d}(B_x+1)(\max\{1,\sqrt{d}B_w\})^L \|\Delta\boldsymbol{\theta}\|_2 \\
&\quad + (\max\{1,\sqrt{d}B_w\})^{L-\ell+1}\Delta d_{\max}^{L+1} \\
&\leq (\max\{1,\sqrt{d}B_w\})^{L-\ell}\cdot \frac{2}{\gamma}L\sqrt{d}(B_x+1)(\max\{1,\sqrt{d}B_w\})^L \|\Delta\boldsymbol{\theta}\|_2 + (\max\{1,\sqrt{d}B_w\})^{L-\ell}\Delta d_{\max}^{L+1} \\
&= (\max\{1,\sqrt{d}B_w\})^{L-\ell}\Big(\frac{2}{\gamma}L\sqrt{d}(B_x+1)(\max\{1,\sqrt{d}B_w\})^L \|\Delta\boldsymbol{\theta}\|_2 + \Delta d_{\max}^{L+1}\Big).
\end{aligned} \tag{67}
$$

Then, our next step is to explicit upper bound $\Delta d_{\max}^{(L+1)}$. By definition, we can write $\Delta d_{\max}^{(L+1)}$ as

$$
\begin{aligned}
\Delta d_{\max}^{(L+1)} &= \max_i \|\mathbf{d}_i^{(L+1)} - \tilde{\mathbf{d}}_i^{(L+1)}\|_2 \\
&= \max_i \left\| \frac{\partial \text{Loss}(f(\mathbf{h}_i^{(L)}), y_i)}{\partial \mathbf{h}_i^{(L)}} - \frac{\partial \text{Loss}(\tilde{f}(\tilde{\mathbf{h}}_i^{(L)}), y_i)}{\partial \tilde{\mathbf{h}}_i^{(L)}} \right\|_2 \\
&\underset{(a)}{\leq} \frac{2}{\gamma}\max_i \left\| \frac{\partial f(\mathbf{h}_i^{(L)})}{\partial \mathbf{h}_i^{(L)}} - \frac{\partial f(\tilde{\mathbf{h}}_i^{(L)})}{\partial \tilde{\mathbf{h}}_i^{(L)}} \right\|_2 \\
&\underset{(b)}{\leq} \frac{2}{\gamma}\Big(\sqrt{d}B_x(\max\{1,\sqrt{d}B_w\})^{L-1}(\|\Delta\mathbf{W}^{(1)}\|_2 + \ldots + \|\Delta\mathbf{W}^{(L)}\|_2) + \big(B_x(\max\{1,\sqrt{d}B_w\})^L + 1\big)\Delta\mathbf{v}\Big) \\
&\leq \frac{2}{\gamma}\Big(\sqrt{d}B_x(\max\{1,\sqrt{d}B_w\})^L + 1\Big)\Big(\|\Delta\mathbf{W}^{(1)}\|_2 + \ldots + \|\Delta\mathbf{W}^{(L)}\|_2 + \|\Delta\mathbf{v}\|_2\Big) \\
&= \frac{2}{\gamma}\Big(\sqrt{d}B_x(\max\{1,\sqrt{d}B_w\})^L + 1\Big)\|\Delta\boldsymbol{\theta}\|_2,
\end{aligned} \tag{68}
$$

where (a) is due to fact that $\nabla\text{Loss}(z,y)$ is $2/\gamma$-Lipschitz continuous with respect to z, (b) follows from Lemma 6 and Lemma 7.

Therefore, we have

$$
\begin{aligned}
\Delta d_{\max}^{(\ell)} &\leq (\max\{1,\sqrt{d}B_w\})^{L-\ell}\Big(\frac{2}{\gamma}L\sqrt{d}(B_x+1)(\max\{1,\sqrt{d}B_w\})^L \|\Delta\boldsymbol{\theta}\|_2 + \Delta d_{\max}^{L+1}\Big) \\
&\leq (\max\{1,\sqrt{d}B_w\})^{L-\ell}\frac{2}{\gamma}\Big(L\sqrt{d}(B_x+1)(\max\{1,\sqrt{d}B_w\})^L + \sqrt{d}B_x(\max\{1,\sqrt{d}B_w\})^L + 1\Big)\|\Delta\boldsymbol{\theta}\|_2 \\
&\leq (\max\{1,\sqrt{d}B_w\})^{L-\ell}\frac{2}{\gamma}\Big((L+1)\sqrt{d}(B_x+1)(\max\{1,\sqrt{d}B_w\})^L + 1\Big)\|\Delta\boldsymbol{\theta}\|_2.
\end{aligned} \tag{69}
$$

F.6 Proof of Lemma 10

By the definition of $\mathbf{G}^{(\ell)},\ \ell\in[L]$, we have

$$
\begin{aligned}
\|\mathbf{G}^{(\ell)}\|_2 &= \frac{1}{m}\|[\mathbf{L}\mathbf{H}^{(\ell-1)}]^\top \mathbf{D}^{(\ell)} \odot \sigma'(\mathbf{Z}^{(\ell)})\|_2 \\
&= \frac{1}{m}\|[\mathbf{L}\mathbf{H}^{(\ell-1)}]^\top \mathbf{D}^{(\ell)}\|_2 \\
&\leq \sqrt{d}h_{\max}^{(\ell-1)} d_{\max}^{(\ell)}.
\end{aligned} \tag{70}
$$

By plugging in the result from Lemma 6 and Lemma 9, we have

$$\|\mathbf{G}^{(\ell)}\|_2 \leq \frac{2}{\gamma}\sqrt{d}(B_x+1)(\max\{1,\sqrt{d}B_w\})^L. \tag{71}$$

Besides, recall that $\mathbf{G}^{(L+1)}$ denotes the gradient with respect to the weight of binary classifier. Therefore, we have

$$\|\mathbf{G}^{(L+1)}\|_2 \leq \frac{2}{\gamma}h_{\max}^{(L)} \leq \frac{2}{\gamma}(\max\{1,\sqrt{d}B_w\})^L B_x, \tag{72}$$

which is smaller than the right hand side of the Eq. 71. Therefore, we have

$$\sum_{\ell=1}^{L+1}\|\mathbf{G}^{(\ell)}\|_2 \leq (L+1)\frac{2}{\gamma}\sqrt{d}(B_x+1)(\max\{1,\sqrt{d}B_w\})^L. \tag{73}$$

Similarly, we can upper bound the difference between gradients computed on two different set of weight parameters for the ℓth layer as

$$\begin{aligned}
\|\mathbf{G}^{(\ell)}-\tilde{\mathbf{G}}^{(\ell)}\|_2 &= \frac{1}{m}\|[\mathbf{L}\mathbf{H}^{(\ell-1)}]^\top\mathbf{D}^{(\ell+1)}\odot\sigma'(\mathbf{Z}^{(\ell)}) - [\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}]^\top\tilde{\mathbf{D}}^{(\ell+1)}\odot\sigma'(\tilde{\mathbf{Z}}^{(\ell)})\|_2 \\
&\leq \frac{1}{m}\|[\mathbf{L}\mathbf{H}^{(\ell-1)}]^\top\mathbf{D}^{(\ell+1)}\odot\sigma'(\mathbf{Z}^{(\ell)}) - [\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}]^\top\mathbf{D}^{(\ell+1)}\odot\sigma'(\mathbf{Z}^{(\ell)})\|_2 \\
&\quad + \frac{1}{m}\|[\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}]^\top\mathbf{D}^{(\ell+1)}\odot\sigma'(\mathbf{Z}^{(\ell)}) - [\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}]^\top\tilde{\mathbf{D}}^{(\ell+1)}\odot\sigma'(\mathbf{Z}^{(\ell)})\|_2 \\
&\quad + \frac{1}{m}\|[\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}]^\top\tilde{\mathbf{D}}^{(\ell+1)}\odot\sigma'(\mathbf{Z}^{(\ell)}) - [\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}]^\top\tilde{\mathbf{D}}^{(\ell+1)}\odot\sigma'(\tilde{\mathbf{Z}}^{(\ell)})\|_2 \\
&\underset{(a)}{\leq} \frac{1}{m}\|[\mathbf{L}(\mathbf{H}^{(\ell-1)}-\tilde{\mathbf{H}}^{(\ell-1)})]^\top\mathbf{D}^{(\ell+1)}\|_2 + \frac{1}{m}\|[\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}]^\top(\mathbf{D}^{(\ell+1)}-\tilde{\mathbf{D}}^{(\ell+1)})\|_2 \\
&\quad + \frac{1}{m}\|[\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}]^\top\tilde{\mathbf{D}}^{(\ell+1)}\odot\big(\sigma'(\mathbf{Z}^{(\ell)})-\sigma'(\tilde{\mathbf{Z}}^{(\ell)})\big)\|_2 \\
&\underset{(b)}{\leq} \max_i\|[[\mathbf{L}(\mathbf{H}^{(\ell-1)}-\tilde{\mathbf{H}}^{(\ell-1)})]^\top\mathbf{D}^{(\ell+1)}]_{i,:}\|_2 \\
&\quad + \max_i\|[[\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}]^\top(\mathbf{D}^{(\ell+1)}-\tilde{\mathbf{D}}^{(\ell+1)})]_{i,:}\|_2 \\
&\quad + \max_i\|[\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}]^\top\tilde{\mathbf{D}}^{(\ell+1)}\|_2\|\mathbf{Z}^{(\ell)}-\tilde{\mathbf{Z}}^{(\ell)}\|_2 \\
&\leq \sqrt{d}\Big(\underbrace{\Delta h_{\max}^{(\ell-1)}d_{\max}^{(\ell+1)}}_{(A)} + \underbrace{h_{\max}^{(\ell-1)}\Delta d_{\max}^{(\ell+1)}}_{(B)} + \underbrace{h_{\max}^{(\ell-1)}d_{\max}^{(\ell+1)}\Delta z_{\max}^{(\ell)}}_{(C)}\Big),
\end{aligned} \tag{74}$$

where inequalities (a) and (b) are due to the fact that the gradient of ReLU is either 0 or 1.

We now proceed to upper bound the three terms on the right hand side. By plugging the result from Lemma 6, Lemma 7, Lemma 9, and letting $C_1 = \max\{1,\sqrt{d}B_w\}$ and $C_2 = \sqrt{d}(B_x+1)$, we can upper bound the term (A) as

$$\begin{aligned}
\Delta h_{\max}^{(\ell-1)}d_{\max}^{(\ell+1)} &\leq \sqrt{d}B_x(\max\{1,\sqrt{d}B_w\})^{\ell-2}\cdot\frac{2}{\gamma}(\max\{1,\sqrt{d}B_w\})^{L-\ell} \\
&\leq \frac{2}{\gamma}\sqrt{d}B_x(\max\{1,\sqrt{d}B_w\})^L\|\Delta\boldsymbol{\theta}\|_2 \\
&\leq \frac{2}{\gamma}C_1^L C_2\|\Delta\boldsymbol{\theta}\|_2,
\end{aligned} \tag{75}$$

upper bound (B) as

$$\begin{aligned}
h_{\max}^{(\ell-1)}\Delta d_{\max}^{(\ell+1)} &\leq B_x(\max\{1,\sqrt{d}B_w\})^L\frac{2}{\gamma}\Big((L+1)\sqrt{d}(B_x+1)(\max\{1,\sqrt{d}B_w\})^L+1\Big)\|\Delta\boldsymbol{\theta}\|_2 \\
&\leq \frac{2}{\gamma}C_1^L C_2\Big((L+1)C_1^L C_2+1\Big)\|\Delta\boldsymbol{\theta}\|_2,
\end{aligned} \tag{76}$$

and upper bound (C) as

$$
\begin{aligned}
h_{\max}^{(\ell-1)} d_{\max}^{(\ell+1)} \Delta z_{\max}^{(\ell)} &\leq B_x(\max\{1, \sqrt{d}B_w\})^{\ell-1} \cdot \frac{2}{\gamma}(\max\{1, \sqrt{d}B_w\})^{L-\ell} \cdot \sqrt{d}B_x(\max\{1, \sqrt{d}B_w\})^{\ell-1}\|\Delta\boldsymbol{\theta}\|_2 \\
&\leq \frac{2}{\gamma}\sqrt{d}B_x^2(\max\{1, \sqrt{d}B_w\})^{2L}\|\Delta\boldsymbol{\theta}\|_2 \\
&\leq \frac{2}{\gamma}(C_1^L C_2)^2\|\Delta\boldsymbol{\theta}\|_2.
\end{aligned} \tag{77}
$$

By combining the results above, we can upper bound $\Delta\mathbf{G}^{(\ell)}$ as

$$
\begin{aligned}
\|\mathbf{G}^{(\ell)} - \tilde{\mathbf{G}}^{(\ell)}\|_2 &\leq \sqrt{d}\Big(\Delta h_{\max}^{(\ell-1)} d_{\max}^{(\ell+1)} + h_{\max}^{(\ell-1)} \Delta d_{\max}^{(\ell+1)} + h_{\max}^{(\ell-1)} d_{\max}^{(\ell+1)} \cdot \Delta z_{\max}^{(\ell)}\Big) \\
&\leq \frac{2}{\gamma} C_1^L C_2 \Big((L+2)C_1^L C_2 + 2\Big)\|\Delta\boldsymbol{\theta}\|_2.
\end{aligned} \tag{78}
$$

Similarly, we can upper bound the difference between gradient for the weight of binary classifier as

$$
\begin{aligned}
\|\mathbf{G}^{(L+1)} - \tilde{\mathbf{G}}^{(L+1)}\|_2 &\leq \frac{2}{\gamma}\Delta h_{\max}^{(L)} \\
&\leq \frac{2}{\gamma}\sqrt{d}B_x(\{1, \sqrt{d}B_w\})^{L-1}\Big(\|\mathbf{W}^{(1)}\|_2 + \ldots + \|\mathbf{W}^{(L)}\|_2\Big),
\end{aligned} \tag{79}
$$

which is smaller than the right hand side of the previous equation. Therefore, we have

$$
\sum_{\ell=1}^{L+1} \|\mathbf{G}^{(\ell)} - \tilde{\mathbf{G}}^{(\ell)}\|_2 \leq (L+1)\frac{2}{\gamma} C_1^L C_2 \Big((L+2)C_1^L C_2 + 2\Big)\|\Delta\boldsymbol{\theta}\|_2. \tag{80}
$$

G Generalization bound for ResGCN

In the following, we provide detailed proof on the generalization bound of ResGCN. Recall that the update rule of ResGCN is defined as

$$
\mathbf{H}^{(\ell)} = \sigma(\mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)}) + \mathbf{H}^{(\ell-1)}, \tag{81}
$$

where $\sigma(\cdot)$ is ReLU activation function. Please notice that although ReLU function $\sigma(x)$ is not differentiable when $x = 0$, for analysis purpose, we suppose the $\sigma'(0) = 0$.

The training of ResGCN is an empirical risk minimization with respect to a set of parameters $\boldsymbol{\theta} = \{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L)}, \mathbf{v}\}$, i.e.,

$$
\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m} \Phi_\gamma(-p(f(\mathbf{h}_i^{(L)}), y_i)),\ f(\mathbf{h}_i^{(L)}) = \tilde{\sigma}(\mathbf{v}^\top \mathbf{h}_i^{(L)}), \tag{82}
$$

where $\mathbf{h}_i^{(L)}$ is the node representation of the ith node at the final layer, $f(\mathbf{h}_i^{(L)})$ is the prediction of the ith node, $\tilde{\sigma}(x) = \frac{1}{\exp(-x)+1}$ is the sigmoid function, and loss function $\Phi_\gamma(-p(z, y))$ is $\frac{2}{\gamma}$-Lipschitz continuous with respect to its first input z. For simplification, we will use $\text{Loss}(z, y)$ denote $\Phi_\gamma(-p(z, y))$ in the proof.

To establish the generalization of ResGCN as stated in Theorem 3, we utilize the result on transductive uniform stability from [16] (Theorem 8 in Appendix F). Then, in Lemma 11, we derive the uniform stability constant for ResGCN, i.e., ϵ_{ResGCN}.

Lemma 11. *The uniform stability constant for ResGCN is computed as* $\epsilon_{\texttt{ResGCN}} = \frac{2\eta\rho_f G_f}{m}\sum_{t=1}^{T}(1 + \eta L_F)^{t-1}$ *where*

$$
\begin{aligned}
&\rho_f = C_1^L C_2,\ G_f = \frac{2}{\gamma}(L+1)C_1^L C_2,\ L_f = \frac{2}{\gamma}(L+1)C_1^L C_2\Big((L+2)C_1^L C_2 + 2\Big), \\
&C_1 = 1 + \sqrt{d}B_w,\ C_2 = \sqrt{d}(B_x + 1).
\end{aligned} \tag{83}
$$

By plugging the result in Lemma 11 back to Theorem 8, we establish the generalization bound for ResGCN.

Similar to the proof on the generalization bound of GCN in Section F, the key idea of the proof is to decompose the change if the ResGCN output into two terms (in Lemma 5) which depend on

- (Lemma 12) The maximum change of node representation, i.e., $\Delta h_{\max}^{(\ell)} = \max_i \|[\mathbf{H}^{(\ell)} - \tilde{\mathbf{H}}^{(\ell)}]_{i,:}\|_2$,
- (Lemma 13) The maximum node representation, i.e., $h_{\max}^{(\ell)} = \max_i \|[\mathbf{H}^{(\ell)}]_{i,:}\|_2$.

Lemma 12 (Upper bound of $h_{\max}^{(\ell)}$ for **ResGCN**). *Let suppose Assumptoon 1 hold. Then, the maximum node embeddings for any node at the ℓth layer is bounded by*

$$h_{\max}^{(\ell)} \leq B_x(1+\sqrt{d}B_w)^{\ell}. \tag{84}$$

Lemma 13 (Upper bound of $\Delta h_{\max}^{(\ell)}$ for **ResGCN**). *Let suppose Assumption 1 hold. Then, the maximum change between the node embeddings on two different set of weight parameters for any node at the ℓth layer is bounded by*

$$\Delta h_{\max}^{(\ell)} \leq \sqrt{d}B_x(1+\sqrt{d}B_w)^{\ell-1}(\|\Delta\mathbf{W}^{(1)}\|_2 + \ldots + \|\Delta\mathbf{W}^{(\ell)}\|_2), \tag{85}$$

where $\Delta\mathbf{W}^{(\ell)} = \mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)}$.

Besides, in Lemma 14, we derive the upper bound on the maximum change of node embeddings before the activation function, i.e., $\Delta z_{\max}^{(\ell)} = \max_i \|[\mathbf{Z}^{(\ell)} - \tilde{\mathbf{Z}}^{(\ell)}]_{i,:}\|_2$, which will be used for the proof of gradient related upper bounds.

Lemma 14 (Upper bound of $\Delta z_{\max}^{(\ell)}$ for **ResGCN**). *Let suppose Assumption 1 hold. Then, the maximum change between the node embeddings before the activation function on two different set of weight parameters for any node at the ℓth layer is bounded by*

$$\Delta z_{\max}^{(\ell)} \leq \sqrt{d}B_x(1+\sqrt{d}B_w)^{\ell-1}(\|\Delta\mathbf{W}^{(1)}\|_2 + \ldots + \|\Delta\mathbf{W}^{(\ell)}\|_2), \tag{86}$$

where $\Delta\mathbf{W}^{(\ell)} = \mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)}$.

Then, in Lemma 15, we decompose the change of the model parameters into two terms which depend on

- The maximum change of gradient passing from the $(\ell+1)$th layer to the ℓth layer

$$\Delta d_{\max}^{(\ell)} = \max_i \left\| \left[\frac{\partial(\sigma(\mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)}) + \mathbf{H}^{(\ell-1)})}{\partial \mathbf{H}^{(\ell-1)}} - \frac{\partial(\sigma(\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}\tilde{\mathbf{W}}^{(\ell)}) + \tilde{\mathbf{H}}^{(\ell-1)})}{\partial \tilde{\mathbf{H}}^{(\ell-1)}} \right]_{i,:} \right\|_2.$$

- The maximum gradient passing from the $(\ell+1)$th layer to the ℓth layer

$$d_{\max}^{(\ell)} = \max_i \left\| \left[\frac{\partial(\sigma(\mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)}) + \mathbf{H}^{(\ell-1)})}{\partial \mathbf{H}^{(\ell-1)}} \right]_{i,:} \right\|_2.$$

Lemma 15 (Upper bound of $d_{\max}^{(\ell)}, \Delta d_{\max}^{(\ell)}$ for **ResGCN**). *Let suppose Assumption 1 hold. Then, the maximum gradient passing from layer $\ell+1$ to layer ℓ for any node is bounded by*

$$d_{\max}^{(\ell)} \leq \frac{2}{\gamma}(1+\sqrt{d}B_w)^{L-\ell+1}, \tag{87}$$

and the maximum change between the gradient passing from layer $\ell+1$ to layer ℓ on two different set of weight parameters for any node is bounded by

$$\Delta d_{\max}^{(\ell)} \leq \frac{2}{\gamma}(1+\sqrt{d}B_w)^{L-\ell}\Big((L+1)\sqrt{d}(B_x+1)(1+\sqrt{d}B_w)^{L}+1\Big)\|\Delta\boldsymbol{\theta}\|_2, \tag{88}$$

where $\|\Delta\boldsymbol{\theta}\|_2 = \|\mathbf{v}-\tilde{\mathbf{v}}\|_2 + \sum_{\ell-1}^{L}\|\mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)}\|_2$ *denotes the change of two set of parameters.*

Finally, based on the previous result, in Lemma 16, we decompose the change of the model parameters into two terms which depend on

- The change of gradient with respect to the ℓth layer weight matrix $\|\Delta \mathbf{G}^{(\ell)}\|_2$,
- The gradient with respect to the ℓth layer weight matrix $\|\mathbf{G}^{(\ell)}\|_2$,

where $\mathbf{G}^{(L+1)}$ denotes the gradient with respect to the weight $\mathbf{v}$ of the binary classifier and $\mathbf{G}^{(\ell)}$ denotes the gradient with respect to the weight $\mathbf{W}^{(\ell)}$ of the ℓth layer graph convolutional layer. Notice that $\|\Delta \mathbf{G}^{(\ell)}\|_2$ reflect the smoothness of ResGCN model and $\|\mathbf{G}^{(\ell)}\|_2$ correspond the upper bound of gradient.

Lemma 16 (Upper bound of $\|\mathbf{G}^{(\ell)}\|_2$, $\|\Delta \mathbf{G}^{(\ell)}\|_2$ for **ResGCN**). *Let suppose Assumption 1 hold and let $C_1 = 1 + \sqrt{d}B_w$ and $C_2 = \sqrt{d}(B_x + 1)$. Then, the gradient and the maximum change between gradients on two different set of weight parameters are bounded by*

$$\begin{aligned} \sum_{\ell=1}^{L+1} \|\mathbf{G}^{(\ell)}\|_2 &\leq \frac{2}{\gamma}(L+1)C_1^L C_2 \|\Delta \boldsymbol{\theta}\|_2, \\ \sum_{\ell=1}^{L+1} \|\Delta \mathbf{G}^{(\ell)}\|_2, &\leq \frac{2}{\gamma}(L+1)C_1^L C_2 \Big((L+2)C_1^L C_2 + 2\Big) \|\Delta \boldsymbol{\theta}\|_2, \end{aligned} \tag{89}$$

where $\|\Delta \boldsymbol{\theta}\|_2 = \|\mathbf{v} - \tilde{\mathbf{v}}\|_2 + \sum_{\ell-1}^{L} \|\mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)}\|_2$ denotes the change of two set of parameters.

Equipped with above intermediate results, we now proceed to prove Lemma 11.

Proof. Recall that our goal is to explore the impact of different GCN structures on the uniform stability constant ϵ_{ResGCN}, which is a function of ρ_f, G_f, and L_f. Let $C_1 = 1 + \sqrt{d}B_w$, $C_2 = \sqrt{d}(B_x + 1)$. Firstly, by plugging Lemma 13 and Lemma 12 into Lemma 5, we have that

$$\begin{aligned} \max_i |f(\mathbf{h}_i^{(L)}) - \tilde{f}(\tilde{\mathbf{h}}_i^{(L)})| &\leq \Delta h_{\max}^{(L)} + h_{\max}^{(L)} \|\Delta \mathbf{v}\|_2 \\ &\leq \sqrt{d}B_x \cdot (\max\{1, \sqrt{d}B_w\})^L \|\Delta \boldsymbol{\theta}\|_2 \\ &\leq C_1^L C_2 \|\Delta \boldsymbol{\theta}\|_2. \end{aligned} \tag{90}$$

Therefore, we know that the function f is ρ_f-Lipschitz continuous, with $\rho_f = C_1^L C_2$. Then, by Lemma 16, we know that the function f is L_f-smoothness, and the gradient of each weight matrices is bounded by G_f, with

$$G_f = \frac{2}{\gamma}(L+1)C_1^L C_2, \; L_f = \frac{2}{\gamma}(L+1)C_1^L C_2 \Big((L+2)C_1^L C_2 + 2\Big). \tag{91}$$

By plugging ϵ_{ResGCN} into Theorem 3, we obtain the generalization bound of ResGCN.

□

G.1 Proof of Lemma 12

By the definition of $h_{\max}^{(\ell)}$, we have

$$\begin{aligned}
h_{\max}^{(\ell)} &= \max_i \|[\sigma(\mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)}) + \mathbf{H}^{(\ell-1)}]_{i,:}\|_2 \\
&\leq \max_i \|[\mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)}]_{i,:}\|_2 + \max_i \|\mathbf{h}_i^{(\ell-1)}\|_2 \\
&\leq \max_i \|[\mathbf{L}\mathbf{H}^{(\ell-1)}]_{i,:}\|_2 \|\mathbf{W}^{(\ell)}\|_2 + h_{\max}^{(\ell-1)} \\
&= \max_i \left\| \sum_{j=1}^N L_{i,j}\mathbf{h}_j^{(\ell-1)} \right\|_2 \|\mathbf{W}^{(\ell)}\|_2 + h_{\max}^{(\ell-1)} \\
&\leq \max_i \left\| \sum_{j=1}^N L_{i,j} \right\|_2 \cdot \max_j \|\mathbf{h}_j^{(\ell-1)}\|_2 \cdot \|\mathbf{W}^{(\ell)}\|_2 + h_{\max}^{(\ell-1)} \\
&\underset{(a)}{\leq} (1 + \sqrt{d}\|\mathbf{W}^{(\ell)}\|_2) \cdot h_{\max}^{(\ell-1)} \\
&\leq (1 + \sqrt{d}B_w) \cdot h_{\max}^{(\ell-1)} \leq B_x(1 + \sqrt{d}B_w)^\ell
\end{aligned} \tag{92}$$

where inequality (a) follows from Lemma 31.

G.2 Proof of Lemma 13

By the definition of $\Delta h_{\max}^{(\ell)}$, we have

$$\begin{aligned}
\Delta h_{\max}^{(\ell)} &= \max_i \|\mathbf{h}_i^{(\ell)} - \tilde{\mathbf{h}}_i^{(\ell)}\|_2 \\
&= \max_i \|[\sigma(\mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)}) - \sigma(\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}\tilde{\mathbf{W}}^{(\ell)}) + \mathbf{H}^{(\ell)} - \tilde{\mathbf{H}}^{(\ell)}]_{i,:}\|_2 \\
&\leq \max_i \|[\mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)} - \mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}\tilde{\mathbf{W}}^{(\ell)}]_{i,:}\|_2 + \max_i \|[\mathbf{H}^{(\ell)} - \tilde{\mathbf{H}}^{(\ell)}]_{i,:}\|_2 \\
&\leq \max_i \|[\mathbf{L}\mathbf{H}^{(\ell-1)}(\mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)}) - \mathbf{L}(\mathbf{H}^{(\ell-1)} - \tilde{\mathbf{H}}^{(\ell-1)})\tilde{\mathbf{W}}^{(\ell)}]_{i,:}\|_2 + \Delta h_{\max}^{(\ell-1)} \\
&\underset{(a)}{\leq} \max_i \left\| \sum_{j=1}^N L_{i,j}\mathbf{h}_j^{(\ell-1)} \right\|_2 \|\Delta\mathbf{W}^{(\ell)}\|_2 + \max_i \left\| \sum_{j=1}^N L_{i,j}\Delta\mathbf{h}_j^{(\ell-1)} \right\|_2 \|\tilde{\mathbf{W}}^{(\ell)}\|_2 + \Delta h_{\max}^{(\ell-1)} \\
&\leq (1 + \sqrt{d}B_w) \cdot \Delta h_{\max}^{(\ell-1)} + \sqrt{d}h_{\max}^{(\ell-1)}\|\Delta\mathbf{W}^{(\ell)}\|_2,
\end{aligned} \tag{93}$$

where inequality (a) follows from Lemma 31.

By induction, we have

$$\begin{aligned}
\Delta h_{\max}^{(\ell)} &\leq (1 + \sqrt{d}B_w) \cdot \Delta h_{\max}^{(\ell-1)} + \sqrt{d}h_{\max}^{(\ell-1)}\|\Delta\mathbf{W}^{(\ell)}\|_2 \\
&\leq (1 + \sqrt{d}B_w)^2 \cdot \Delta h_{\max}^{(\ell-2)} \\
&\quad + \sqrt{d}\Big(h_{\max}^{(\ell-1)}\|\Delta\mathbf{W}^{(\ell)}\|_2 + (1 + \sqrt{d}B_w)h_{\max}^{(\ell-2)}\|\Delta\mathbf{W}^{(\ell-1)}\|_2\Big) \\
&\ldots \\
&\leq (1 + \sqrt{d}B_w)^\ell \cdot \Delta h_{\max}^{(0)} \\
&\quad + \sqrt{d}\Big(h_{\max}^{(\ell-1)}\|\Delta\mathbf{W}^{(\ell)}\|_2 + (1 + \sqrt{d}B_w)h_{\max}^{(\ell-2)}\|\Delta\mathbf{W}^{(\ell-1)}\|_2 + \ldots + (1 + \sqrt{d}B_w)^{\ell-1}h_{\max}^{(0)}\|\Delta\mathbf{W}^{(1)}\|_2\Big) \\
&= \sqrt{d}\Big(h_{\max}^{(\ell-1)}\|\Delta\mathbf{W}^{(\ell)}\|_2 + (1 + \sqrt{d}B_w)h_{\max}^{(\ell-2)}\|\Delta\mathbf{W}^{(\ell-1)}\|_2 + \ldots + (1 + \sqrt{d}B_w)^{\ell-1}h_{\max}^{(0)}\|\Delta\mathbf{W}^{(1)}\|_2\Big),
\end{aligned} \tag{94}$$

where the last equality is due to $\Delta h_{\max}^{(0)} = 0$.

Plugging in the upper bound of $h_{\max}^{(\ell)}$ in Lemma 13, we have

$$\Delta h_{\max}^{(\ell)} \leq \sqrt{d}B_x(1 + \sqrt{d}B_w)^{\ell-1}(\|\mathbf{W}^{(1)}\|_2 + \ldots + \|\mathbf{W}^{(\ell)}\|_2). \tag{95}$$

G.3 Proof of Lemma 14

By the definition of $\mathbf{Z}^{(\ell)}$, we have

$$\begin{aligned}\mathbf{Z}^{(\ell)} - \tilde{\mathbf{Z}}^{(\ell)} &= \mathbf{L}\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)} - \mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}\tilde{\mathbf{W}}^{(\ell)} \\ &= \mathbf{L}\Big(\mathbf{H}^{(\ell-1)}\mathbf{W}^{(\ell)} - \tilde{\mathbf{H}}^{(\ell-1)}\mathbf{W}^{(\ell)} + \tilde{\mathbf{H}}^{(\ell-1)}\mathbf{W}^{(\ell)} - \tilde{\mathbf{H}}^{(\ell-1)}\tilde{\mathbf{W}}^{(\ell)}\Big) \\ &= \mathbf{L}\big(\mathbf{H}^{(\ell-1)} - \tilde{\mathbf{H}}^{(\ell-1)}\big)\mathbf{W}^{(\ell)} + \mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}\big(\mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)}\big).\end{aligned} \tag{96}$$

Then, by taking the norm of both sides, we have

$$\begin{aligned}\Delta z_{\max}^{(\ell)} &= \max_i \|[\mathbf{Z}^{(\ell)} - \tilde{\mathbf{Z}}^{(\ell)}]_{i,:}\|_2 \\ &\le \sqrt{d}B_w \cdot \max_i \|[\mathbf{Z}^{(\ell-1)} - \tilde{\mathbf{Z}}^{(\ell-1)}]_{i,:}\|_2 + \sqrt{d}h_{\max}^{(\ell-1)}\|\Delta\mathbf{W}^{(\ell)}\|_2 \\ &= \sqrt{d}B_w \cdot z_{\max}^{(\ell-1)} + \sqrt{d}h_{\max}^{(\ell-1)}\|\Delta\mathbf{W}^{(\ell)}\|_2.\end{aligned} \tag{97}$$

By induction, we have

$$\begin{aligned}\Delta z_{\max}^{(\ell)} &\le \sqrt{d}B_w \cdot \Delta z_{\max}^{(\ell-1)} + \sqrt{d}h_{\max}^{(\ell-1)}\|\Delta\mathbf{W}^{(\ell)}\|_2 \\ &\underset{(a)}{\le} \sqrt{d}B_w \cdot \Delta z_{\max}^{(\ell-1)} + \sqrt{d}B_x(\sqrt{d}B_w)^{\ell-1}\|\Delta\mathbf{W}^{(\ell)}\|_2 \\ &\le \sqrt{d}B_w\Big(\sqrt{d}B_w \cdot \Delta z_{\max}^{(\ell-2)} + \sqrt{d}B_x(\sqrt{d}B_w)^{\ell-2}\|\Delta\mathbf{W}^{(\ell-1)}\|_2\Big) + \sqrt{d}B_x(1+\sqrt{d}B_w)^{\ell-1}\|\Delta\mathbf{W}^{(\ell)}\|_2 \\ &= (\sqrt{d}B_w)^2 \cdot \Delta z_{\max}^{(\ell-2)} + \sqrt{d}B_x(\sqrt{d}B_w)^{\ell-1}\Big(\|\Delta\mathbf{W}^{(\ell-1)}\|_2 + \|\Delta\mathbf{W}^{(\ell)}\|_2\Big) \\ &\le \dots \\ &\le \sqrt{d}B_x(\sqrt{d}B_w)^{\ell-1}\Big(\|\Delta\mathbf{W}^{(1)}\|_2 + \dots + \|\Delta\mathbf{W}^{(\ell)}\|_2\Big),\end{aligned} \tag{98}$$

where (a) is due to $h_{\max}^{(\ell-1)} \le (\sqrt{d}B_w)^{\ell-1}B_x$.

G.4 Proof of Lemma 15

For notation simplicity, let $\mathbf{D}^{(\ell)}$ denote the gradient passing from the ℓth to the $(\ell-1)$th layer. By the definition of $d_{\max}^{(\ell)}$, we have

$$\begin{aligned}d_{\max}^{(\ell)} &= \max_i \|\mathbf{d}_i^{(\ell)}\|_2 \\ &= \max_i [\|\mathbf{L}^\top \sigma'(\mathbf{Z}^{(\ell)}) \odot \mathbf{D}^{(\ell+1)}\mathbf{W}^{(\ell)} + \mathbf{D}^{(\ell+1)}\|]_{i,:} \\ &\le \max_i [\|\mathbf{L}^\top \mathbf{D}^{(\ell+1)}\|_2]_{i,:}\|\mathbf{W}^{(\ell)}\|_2 + d_{\max}^{(\ell+1)} \\ &\le \max_i \left\|\sum_{j=1}^N L_{i,j}^\top \mathbf{d}_j^{(\ell+1)}\right\|_2 \|\mathbf{W}^{(\ell)}\|_2 + d_{\max}^{(\ell+1)} \\ &\le (1+\sqrt{d}\|\mathbf{W}^{(\ell)}\|_2)d_{\max}^{(\ell+1)} \\ &\underset{(a)}{\le} (1+\sqrt{d}B_w)\cdot d_{\max}^{(\ell+1)} \le \frac{2}{\gamma}(1+\sqrt{d}B_w)^{L-\ell+1},\end{aligned} \tag{99}$$

where inequality (a) follows from Lemma 31.

By the definition of $\Delta d_{\max}^{(\ell)}$, we have

$$
\begin{aligned}
\Delta d_{\max}^{(\ell)} &= \max_i \|[\mathbf{D}^{(\ell)} - \tilde{\mathbf{D}}^{(\ell)}]_{i,:}\|_2 \\
&= \max_i \|[\mathbf{L}^\top(\mathbf{D}^{(\ell+1)} \odot \sigma'(\mathbf{Z}^{(\ell)}))[\mathbf{W}^{(\ell)}]^\top + \mathbf{D}^{(\ell+1)} - \mathbf{L}^\top(\tilde{\mathbf{D}}^{(\ell+1)} \odot \sigma'(\tilde{\mathbf{Z}}^{(\ell)}))[\tilde{\mathbf{W}}^{(\ell)}]^\top - \tilde{\mathbf{D}}^{(\ell+1)}]_{i,:}\|_2 \\
&\leq \max_i \|[\mathbf{L}^\top(\mathbf{D}^{(\ell+1)} \odot \sigma'(\mathbf{Z}^{(\ell)}))[\mathbf{W}^{(\ell)}]^\top - \mathbf{L}^\top(\tilde{\mathbf{D}}^{(\ell+1)} \odot \sigma'(\mathbf{Z}^{(\ell)}))[\mathbf{W}^{(\ell)}]^\top]_{i,:}\|_2 \\
&\quad + \max_i \|[\mathbf{L}^\top(\tilde{\mathbf{D}}^{(\ell+1)} \odot \sigma'(\mathbf{Z}^{(\ell)}))[\mathbf{W}^{(\ell)}]^\top - \mathbf{L}^\top(\tilde{\mathbf{D}}^{(\ell+1)} \odot \sigma'(\mathbf{Z}^{(\ell)}))[\tilde{\mathbf{W}}^{(\ell)}]^\top]_{i,:}\|_2 \\
&\quad + \max_i \|[\mathbf{L}^\top(\tilde{\mathbf{D}}^{(\ell+1)} \odot \sigma'(\mathbf{Z}^{(\ell)}))[\tilde{\mathbf{W}}^{(\ell)}]^\top - \mathbf{L}^\top(\tilde{\mathbf{D}}^{(\ell+1)} \odot \sigma'(\tilde{\mathbf{Z}}^{(\ell)}))[\tilde{\mathbf{W}}^{(\ell)}]^\top]_{i,:}\|_2 + \max_i \|[\mathbf{D}^{(\ell+1)} - \tilde{\mathbf{D}}^{(\ell+1)}]_{i,:}\|_2 \\
&\underset{(a)}{\leq} \max_i \|[\mathbf{L}^\top((\mathbf{D}^{(\ell+1)} - \tilde{\mathbf{D}}^{(\ell+1)}))[\mathbf{W}^{(\ell)}]^\top]_{i,:}\|_2 + \max_i \|[\mathbf{L}^\top(\tilde{\mathbf{D}}^{(\ell+1)}[\mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)}]^\top]_{i,:}\|_2 \\
&\quad + \max_i \|[\mathbf{L}^\top\tilde{\mathbf{D}}^{(\ell+1)}[\tilde{\mathbf{W}}^{(\ell)}]^\top]_{i,:}\|_2 \max_i \|[\mathbf{Z}^{(\ell)} - \tilde{\mathbf{Z}}^{(\ell)}]_{i,:}\| + \max_i \|[\mathbf{D}^{(\ell+1)} - \tilde{\mathbf{D}}^{(\ell+1)}]_{i,:}\|_2 \\
&\leq (1 + \sqrt{d}B_w)\Delta d_{\max}^{(\ell+1)} + \sqrt{d}d_{\max}^{(\ell+1)}\|\Delta\mathbf{W}^{(\ell)}\|_2 + \sqrt{d}B_w d_{\max}^{(\ell+1)}\Delta z_{\max}^{(\ell)} \\
&= (1 + \sqrt{d}B_w)\Delta d_{\max}^{(\ell+1)} + \underbrace{d_{\max}^{(\ell+1)}\Big(\sqrt{d}\|\Delta\mathbf{W}^{(\ell)}\|_2 + \sqrt{d}B_w\Delta z_{\max}^{(\ell)}\Big)}_{(A)},
\end{aligned}
\tag{100}
$$

where inequality (a) is due to the fact that the gradient of ReLU is either 1 or 0.

Knowing that $d_{\max}^{(\ell+1)} \leq \frac{2}{\gamma}(1 + \sqrt{d}B_w)^{L-\ell}$ and $z_{\max}^{(\ell)} \leq \sqrt{d}B_x(\sqrt{d}B_w)^{\ell-1}\Big(\|\mathbf{W}^{(1)}\|_2 + \ldots + \|\mathbf{W}^{(\ell)}\|_2\Big)$, we can upper bound (A) by

$$
\begin{aligned}
& d_{\max}^{(\ell+1)}\Big(\sqrt{d}\|\Delta\mathbf{W}^{(\ell)}\|_2 + \sqrt{d}B_w\Delta z_{\max}^{(\ell)}\Big) \\
&\leq \frac{2}{\gamma}(1 + \sqrt{d}B_w)^{L-\ell}\Big(\sqrt{d}\|\Delta\mathbf{W}^{(\ell)}\|_2 + \sqrt{d}B_x(\sqrt{d}B_w)^{\ell}\big(\|\mathbf{W}^{(1)}\|_2 + \ldots + \|\mathbf{W}^{(\ell)}\|_2\big)\Big) \\
&\leq \frac{2}{\gamma}(1 + \sqrt{d}B_w)^{L-\ell}\Big(\sqrt{d} + \sqrt{d}B_x(\sqrt{d}B_w)^{\ell}\Big)\big(\|\mathbf{W}^{(1)}\|_2 + \ldots + \|\mathbf{W}^{(\ell)}\|_2\big) \\
&\leq \frac{2}{\gamma}\sqrt{d}(1 + B_x)(1 + \sqrt{d}B_w)^{L}\|\Delta\boldsymbol{\theta}\|_2.
\end{aligned}
\tag{101}
$$

By plugging it back, we have

$$
\begin{aligned}
\Delta d_{\max}^{(\ell)} &\leq \sqrt{d}B_w\Delta d_{\max}^{(\ell+1)} + \frac{2}{\gamma}\sqrt{d}(B_x + 1)(1 + \sqrt{d}B_w)^{L}\|\Delta\boldsymbol{\theta}\|_2 \\
&\leq \sqrt{d}B_w\Big(\sqrt{d}B_w\Delta d_{\max}^{(\ell+2)} + \frac{2}{\gamma}\sqrt{d}(B_x + 1)(1 + \sqrt{d}B_w)^{L}\|\Delta\boldsymbol{\theta}\|_2\Big) \\
&\quad + \frac{2}{\gamma}\sqrt{d}(B_x + 1)(1 + \sqrt{d}B_w)^{L}\|\Delta\boldsymbol{\theta}\|_2 \\
&\leq \Big(1 + \sqrt{d}B_w + \ldots + (1 + \sqrt{d}B_w)^{L-\ell-1}\Big) \cdot \frac{2}{\gamma}\sqrt{d}(B_x + 1)(1 + \sqrt{d}B_w)^{L}\|\Delta\boldsymbol{\theta}\|_2 \\
&\quad + (1 + \sqrt{d}B_w)^{L-\ell+1}\Delta d_{\max}^{L+1} \\
&\leq (1 + \sqrt{d}B_w)^{L-\ell} \cdot \frac{2}{\gamma}L\sqrt{d}(B_x + 1)(1 + \sqrt{d}B_w)^{L}\|\Delta\boldsymbol{\theta}\|_2 + (1 + \sqrt{d}B_w)^{L-\ell}\Delta d_{\max}^{L+1} \\
&= (1 + \sqrt{d}B_w)^{L-\ell}\Big(\frac{2}{\gamma}L\sqrt{d}(B_x + 1)(1 + \sqrt{d}B_w)^{L}\|\Delta\boldsymbol{\theta}\|_2 + \Delta d_{\max}^{L+1}\Big).
\end{aligned}
\tag{102}
$$

Let first explicit upper bound $\Delta d_{\max}^{(L+1)}$. By definition, we can write $\Delta d_{\max}^{(L+1)}$ as

$$\begin{aligned}
\Delta d_{\max}^{(L+1)} &= \max_i \|\mathbf{d}_i^{(L+1)} - \tilde{\mathbf{d}}_i^{(L+1)}\|_2 \\
&= \max_i \left\| \frac{\partial \text{Loss}(f(\mathbf{h}_i^{(L)}), y_i)}{\partial \mathbf{h}_i^{(L)}} - \frac{\partial \text{Loss}(\tilde{f}(\tilde{\mathbf{h}}_i^{(L)}), y_i)}{\partial \tilde{\mathbf{h}}_i^{(L)}} \right\|_2 \\
&\underset{(a)}{\leq} \frac{2}{\gamma} \max_i \left\| \frac{\partial f(\mathbf{h}_i^{(L)})}{\partial \mathbf{h}_i^{(L)}} - \frac{\partial f(\tilde{\mathbf{h}}_i^{(L)})}{\partial \tilde{\mathbf{h}}_i^{(L)}} \right\|_2 \\
&\underset{(b)}{\leq} \frac{2}{\gamma} \Big(\sqrt{d} B_x (1 + \sqrt{d} B_w)^{L-1} (\|\Delta \mathbf{W}^{(1)}\|_2 + \ldots + \|\Delta \mathbf{W}^{(L)}\|_2) + \big(B_x (1 + \sqrt{d} B_w)^L + 1 \big) \Delta \mathbf{v} \Big) \\
&\leq \frac{2}{\gamma} \Big(\sqrt{d} B_x (1 + \sqrt{d} B_w)^L + 1 \Big) \Big(\|\Delta \mathbf{W}^{(1)}\|_2 + \ldots + \|\Delta \mathbf{W}^{(L)}\|_2 + \|\Delta \mathbf{v}\|_2 \Big) \\
&= \frac{2}{\gamma} \Big(\sqrt{d} B_x (1 + \sqrt{d} B_w)^L + 1 \Big) \|\Delta \boldsymbol{\theta}\|_2,
\end{aligned} \tag{103}$$

where inequality (a) is due to the fact that $\nabla \text{Loss}(z, y)$ is $\frac{2}{\gamma}$-Lipschitz continuous with respect to z, and inequality (b) follows from Lemma 12 and Lemma 13. Therefore, we have

$$\begin{aligned}
\Delta d_{\max}^{(\ell)} &\leq (1 + \sqrt{d} B_w)^{L-\ell} \Big(\frac{2}{\gamma} L \sqrt{d} (B_x + 1)(1 + \sqrt{d} B_w)^L \|\Delta \boldsymbol{\theta}\|_2 + \Delta d_{\max}^{L+1} \Big) \\
&\leq (1 + \sqrt{d} B_w)^{L-\ell} \frac{2}{\gamma} \Big(L \sqrt{d} (B_x + 1)(1 + \sqrt{d} B_w)^L + \sqrt{d} B_x (1 + \sqrt{d} B_w)^L + 1 \Big) \|\Delta \boldsymbol{\theta}\|_2 \\
&\leq (1 + \sqrt{d} B_w)^{L-\ell} \frac{2}{\gamma} \Big((L+1) \sqrt{d} (B_x + 1)(1 + \sqrt{d} B_w)^L + 1 \Big) \|\Delta \boldsymbol{\theta}\|_2.
\end{aligned} \tag{104}$$

G.5 Proof of Lemma 16

First By the definition of $\mathbf{G}^{(\ell)}$, $\ell \in [L]$, we have

$$\begin{aligned}
\|\mathbf{G}^{(\ell)}\|_2 &= \frac{1}{m} \|[\mathbf{L}\mathbf{H}^{(\ell-1)}]^\top \mathbf{D}^{(\ell)} \odot \sigma'(\boldsymbol{Z}^{(\ell)})\|_2 \\
&\leq \frac{1}{m} \|[\mathbf{L}\mathbf{H}^{(\ell-1)}]^\top \mathbf{D}^{(\ell)}\|_2 \\
&\leq \sqrt{d} h_{\max}^{(\ell-1)} d_{\max}^{(\ell)} \\
&\underset{(a)}{\leq} \frac{2}{\gamma} \sqrt{d} (B_x + 1)(1 + \sqrt{d} B_w)^L,
\end{aligned} \tag{105}$$

where inequality (a) follows from Lemma 12 and Lemma 15, and the fact that loss function is $\frac{2}{\gamma}$-Lipschitz continuous.

Similarly, by the definition of $\mathbf{G}^{(L+1)}$, we have

$$\|\mathbf{G}^{(L+1)}\|_2 \underset{(a)}{\leq} \frac{2}{\gamma} h_{\max}^{(L)} \underset{(b)}{\leq} \frac{2}{\gamma} (B_x + 1)(1 + \sqrt{d} B_w)^L, \tag{106}$$

which is smaller then the right hind side of Eq. 105, and inequality (a) is due to the fact that loss function is $\frac{2}{\gamma}$-Lipschitz continuous and inequality (b) follows from Lemma 12.

By combining the above two inequalities together, we have

$$\sum_{\ell=1}^{L+1} \|\mathbf{G}^{(\ell)}\|_2 \leq \frac{2}{\gamma} (L+1) \sqrt{d} (B_x + 1)(1 + \sqrt{d} B_w)^L. \tag{107}$$

Furthermore, we can upper bound the difference between gradients computed on two different set of weight parameters for the ℓth layer as

$$
\begin{aligned}
\|\mathbf{G}^{(\ell)} - \tilde{\mathbf{G}}^{(\ell)}\|_2 &= \frac{1}{m}\|[\mathbf{L}\mathbf{H}^{(\ell-1)}]^\top \mathbf{D}^{(\ell+1)} \odot \sigma'(\mathbf{Z}^{(\ell)}) - [\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}]^\top \tilde{\mathbf{D}}^{(\ell+1)} \odot \sigma'(\tilde{\mathbf{Z}}^{(\ell)})\|_2 \\
&\leq \frac{1}{m}\|[\mathbf{L}\mathbf{H}^{(\ell-1)}]^\top \mathbf{D}^{(\ell+1)} \odot \sigma'(\mathbf{Z}^{(\ell)}) - [\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}]^\top \mathbf{D}^{(\ell+1)} \odot \sigma'(\mathbf{Z}^{(\ell)})\|_2 \\
&\quad + \frac{1}{m}\|[\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}]^\top \mathbf{D}^{(\ell+1)} \odot \sigma'(\mathbf{Z}^{(\ell)}) - [\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}]^\top \tilde{\mathbf{D}}^{(\ell+1)} \odot \sigma'(\mathbf{Z}^{(\ell)})\|_2 \\
&\quad + \frac{1}{m}\|[\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}]^\top \tilde{\mathbf{D}}^{(\ell+1)} \odot \sigma'(\mathbf{Z}^{(\ell)}) - [\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}]^\top \tilde{\mathbf{D}}^{(\ell+1)} \odot \sigma'(\tilde{\mathbf{Z}}^{(\ell)})\|_2 \\
&\underset{(a)}{\leq} \frac{1}{m}\|[\mathbf{L}(\mathbf{H}^{(\ell-1)} - \tilde{\mathbf{H}}^{(\ell-1)})]^\top \mathbf{D}^{(\ell+1)}\|_2 + \frac{1}{m}\|[\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}]^\top (\mathbf{D}^{(\ell+1)} - \tilde{\mathbf{D}}^{(\ell+1)})\|_2 \\
&\quad + \frac{1}{m}\|[\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}]^\top \tilde{\mathbf{D}}^{(\ell+1)} \odot \big(\sigma'(\mathbf{Z}^{(\ell)}) - \sigma'(\tilde{\mathbf{Z}}^{(\ell)})\big)\|_2 \\
&\underset{(b)}{\leq} \max_i \|[[\mathbf{L}(\mathbf{H}^{(\ell-1)} - \tilde{\mathbf{H}}^{(\ell-1)})]^\top \mathbf{D}^{(\ell+1)}]_{i,:}\|_2 \\
&\quad + \max_i \|[[\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}]^\top (\mathbf{D}^{(\ell+1)} - \tilde{\mathbf{D}}^{(\ell+1)})]_{i,:}\|_2 \\
&\quad + \max_i \|[\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}]^\top \tilde{\mathbf{D}}^{(\ell+1)}\|_2 \|\mathbf{Z}^{(\ell)} - \tilde{\mathbf{Z}}^{(\ell)}\|_2 \\
&\leq \sqrt{d}\Big(\underbrace{\Delta h_{\max}^{(\ell-1)} d_{\max}^{(\ell+1)}}_{(A)} + \underbrace{h_{\max}^{(\ell-1)} \Delta d_{\max}^{(\ell+1)}}_{(B)} + \underbrace{h_{\max}^{(\ell-1)} d_{\max}^{(\ell+1)} \Delta z_{\max}^{(\ell)}}_{(C)}\Big),
\end{aligned}
\tag{108}
$$

where inequality (a) and (b) is due to the fact that the gradient of ReLU activation function is element-wise either 0 or 1.

By plugging the result from Lemma 12, Lemma 13, Lemma 15, and letting $C_1 = 1 + \sqrt{d}B_w$ and $C_2 = \sqrt{d}(B_x + 1)$ we can upper bound (A) as

$$
\begin{aligned}
\Delta h_{\max}^{(\ell-1)} d_{\max}^{(\ell+1)} &\leq \sqrt{d}B_x(1 + \sqrt{d}B_w)^{\ell-2} \cdot \frac{2}{\gamma}(1 + \sqrt{d}B_w)^{L-\ell} \\
&\leq \frac{2}{\gamma}\sqrt{d}B_x(1 + \sqrt{d}B_w)^{L}\|\Delta\boldsymbol{\theta}\|_2 \\
&\leq \frac{2}{\gamma}C_1^L C_2\|\Delta\boldsymbol{\theta}\|_2,
\end{aligned}
\tag{109}
$$

upper bound (B) as

$$
\begin{aligned}
h_{\max}^{(\ell-1)} \Delta d_{\max}^{(\ell+1)} &\leq B_x(\max\{1, \sqrt{d}B_w\})^L \frac{2}{\gamma}\Big((L+1)\sqrt{d}(B_x+1)(\max\{1, \sqrt{d}B_w\})^L + 1\Big)\|\Delta\boldsymbol{\theta}\|_2 \\
&\leq \frac{2}{\gamma}C_1^L C_2\Big((L+1)C_1^L C_2 + 1\Big)\|\Delta\boldsymbol{\theta}\|_2,
\end{aligned}
\tag{110}
$$

and upper bound (C) as

$$
\begin{aligned}
h_{\max}^{(\ell-1)} d_{\max}^{(\ell+1)} \Delta z_{\max}^{(\ell)} &\leq B_x(1 + \sqrt{d}B_w)^{\ell-1} \cdot \frac{2}{\gamma}(1 + \sqrt{d}B_w)^{L-\ell} \cdot \sqrt{d}B_x(1 + \sqrt{d}B_w)^{\ell-1}\|\Delta\boldsymbol{\theta}\|_2 \\
&\leq \frac{2}{\gamma}\sqrt{d}B_x^2(1 + \sqrt{d}B_w)^{2L}\|\Delta\boldsymbol{\theta}\|_2 \\
&\leq \frac{2}{\gamma}(C_1^L C_2)^2\|\Delta\boldsymbol{\theta}\|_2.
\end{aligned}
\tag{111}
$$

By combining the results above, we can upper bound $\Delta\mathbf{G}^{(\ell)}$ as

$$
\begin{aligned}
\|\mathbf{G}^{(\ell)} - \tilde{\mathbf{G}}^{(\ell)}\|_2 &\leq \sqrt{d}\Big(\Delta h_{\max}^{(\ell-1)} d_{\max}^{(\ell+1)} + h_{\max}^{(\ell-1)} \Delta d_{\max}^{(\ell+1)} + h_{\max}^{(\ell-1)} d_{\max}^{(\ell+1)} \cdot \Delta z_{\max}^{(\ell)}\Big) \\
&\leq \frac{2}{\gamma}C_1^L C_2\Big((L+2)C_1^L C_2 + 2\Big)\|\Delta\boldsymbol{\theta}\|_2.
\end{aligned}
\tag{112}
$$

By plugging the result from Lemma 12, Lemma 13, Lemma 15, we can upper bound $\Delta\mathbf{G}^{(\ell)}$ as

$$\begin{aligned}\|\Delta\mathbf{G}^{(\ell)}\|_2 &\leq \sqrt{d}\Big(\Delta h_{\max}^{(\ell-1)} d_{\max}^{(\ell+1)} + h_{\max}^{(\ell-1)}\Delta d_{\max}^{(\ell+1)}\Big) \\ &\leq \frac{2}{\gamma}\Big(\sqrt{d}B_x(1+\sqrt{d}B_w)^L + (1+\sqrt{d}B_w) + \sqrt{d}\Big)\|\Delta\boldsymbol{\theta}\|_2 \cdot \sqrt{d}B_x(1+\sqrt{d}B_w)^L.\end{aligned} \tag{113}$$

Similarly, we can upper bound the difference between gradient for the weight parameters of the binary classifier as

$$\begin{aligned}\|\mathbf{G}^{(L+1)} - \tilde{\mathbf{G}}^{(L+1)}\|_2 &\underset{(a)}{\leq} \frac{2}{\gamma}\Delta h_{\max}^{(L)} \\ &\leq \frac{2}{\gamma}\sqrt{d}B_x(1+\sqrt{d}B_w)^{L-1}(\|\mathbf{W}^{(1)}\|_2 + \ldots + \|\mathbf{W}^{(L)}\|_2),\end{aligned} \tag{114}$$

where (a) is due to the fact that loss function is $\frac{2}{\gamma}$-Lipschitz continuous.

Therefore, by combining the above inequalites, we have

$$\sum_{\ell=1}^{L}\|\Delta\mathbf{G}^{(\ell)}\|_2 \leq \frac{2}{\gamma}(L+1)\Big(\sqrt{d}B_x(1+\sqrt{d}B_w)^L + (1+\sqrt{d}B_w) + \sqrt{d}\Big)\|\Delta\boldsymbol{\theta}\|_2 \cdot \sqrt{d}B_x(1+\sqrt{d}B_w)^L. \tag{115}$$

H Generalization bound for APPNP

In the following, we provide detailed proof on the generalization bound of APPNP. Recall that the update rule of APPNP is defined as

$$\mathbf{H}^{(\ell)} = \alpha\mathbf{L}\mathbf{H}^{(\ell-1)} + (1-\alpha)\mathbf{H}^{(0)},\ \mathbf{H}^{(0)} = \mathbf{W}\mathbf{X}. \tag{116}$$

The training of APPNP is an empirical risk minimization with respect to weight parameters $\boldsymbol{\theta} = \{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L)}, \boldsymbol{v}\}$, i.e.,

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m}\Phi_\gamma(-p(f(\mathbf{h}_i^{(L)}), y_i)),\ f(\mathbf{h}_i^{(L)}) = \tilde{\sigma}(\boldsymbol{v}^\top\mathbf{h}_i^{(L)}), \tag{117}$$

where $\mathbf{h}_i^{(L)}$ is the node representation of the ith node at the final layer, $f(\mathbf{h}_i^{(L)})$ is the predicted label for the ith node, $\tilde{\sigma}(x) = \frac{1}{\exp(-x)+1}$ is the sigmoid function, and loss function $\Phi_\gamma(-p(z,y))$ is $\frac{2}{\gamma}$-Lipschitz continuous with respect to its first input z. For simplification, we will use $\text{Loss}(z,y)$ denote $\Phi_\gamma(-p(z,y))$ in the proof.

To establish the generalization of APPNP as stated in Theorem 3, we utilize the result on transductive uniform stability from [16] (Theorem 8 in Appendix F). Then, in Lemma 17, we derive the uniform stability constant for APPNP, i.e., ϵ_{APPNP}.

Lemma 17. *The uniform stability constant for APPNP is computed as* $\epsilon_{\texttt{APPNP}} = \frac{2\eta\rho_f G_f}{m}\sum_{t=1}^{T}(1+\eta L_F)^{t-1}$ *where*

$$\begin{aligned}&\rho_f = C_1 B_w,\ G_f = \frac{4}{\gamma}C_1, L_f\ = \frac{4}{\gamma}C_1\big(C_1C_2+1\big), \\ &C_1 = B_d^\alpha B_x,\ C_2 = \max\{1, B_w\},\ B_d^\alpha = (1-\alpha)\sum_{\ell=1}^{L}(\alpha\sqrt{d})^{\ell-1} + (\alpha\sqrt{d})^L.\end{aligned} \tag{118}$$

By plugging the result in Lemma 17 back to Theorem 8, we establish the generalization bound for APPNP.

The key idea of the proof is to decompose the change of the APPNP output into two terms (in Lemma 5) which depend on

- (Lemma 18) The maximum change of node representation $\Delta h_{\max}^{(L)} = \max_i \|[\mathbf{H}^{(L)} - \tilde{\mathbf{H}}^{(L)}]_{i,:}\|_2$,
- (Lemma 19) The maximum node representation $h_{\max}^{(L)} = \max_i \|[\mathbf{H}^{(L)}]_{i,:}\|_2$.

Lemma 18 (Upper bound of $h_{\max}^{(L)}$ for **APPNP**). *Let suppose Assumption 1 hold. Then, the maximum node embeddings for any node at the ℓth layer is bounded by*

$$h_{\max}^{(L)} \leq B_d^\alpha B_x B_w, \tag{119}$$

where $B_d^\alpha = (1-\alpha)\sum_{\ell=1}^{L}(\alpha\sqrt{d})^{\ell-1} + (\alpha\sqrt{d})^L$.

Lemma 19 (Upper bound of $\Delta h_{\max}^{(L)}$ for **APPNP**). *Let suppose Assumption 1 hold. Then, the maximum change between the node embeddings on two different set of weight parameters for any node at the ℓth layer is bounded by*

$$\Delta h_{\max}^{(L)} \leq B_d^\alpha B_x \|\Delta\mathbf{W}\|_2, \tag{120}$$

where $B_d^\alpha = (1-\alpha)\sum_{\ell=1}^{L}(\alpha\sqrt{d})^{\ell-1} + (\alpha\sqrt{d})^L$ *and* $\Delta\mathbf{W}^{(\ell)} = \mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)}$.

Then, in Lemma 20, we decompose the change of the model parameters into two terms which depend on

- The change of gradient with respect to the ℓth layer weight matrix $\|\Delta\mathbf{G}^{(\ell)}\|_2$,
- The gradient with respect to the ℓth layer weight matrix $\|\mathbf{G}^{(\ell)}\|_2$,

where $\|\Delta\mathbf{G}^{(\ell)}\|_2$ reflects the smoothness of APPNP model and $\|\mathbf{G}^{(\ell)}\|_2$ corresponds the upper bound of gradient.

Lemma 20 (Upper bound of $\|\mathbf{G}^{(\ell)}\|_2$, $\|\Delta\mathbf{G}^{(\ell)}\|_2$ for **APPNP**). *Let suppose Assumption 1 hold. Then, the gradient and the maximum change between gradients on two different set of weight parameters are bounded by*

$$\begin{aligned}
\sum_{\ell=1}^{2}\|\mathbf{G}^{(\ell)}\|_2 &\leq \frac{4}{\gamma}B_d^\alpha B_x, \\
\sum_{\ell=1}^{2}\|\Delta\mathbf{G}^{(1)}\|_2 &\leq \frac{4}{\gamma}B_d^\alpha B_x\Big(B_d^\alpha B_x \max\{1, B_w\} + 1\Big)\|\Delta\boldsymbol{\theta}\|_2,
\end{aligned} \tag{121}$$

where $\|\Delta\boldsymbol{\theta}\|_2 = \|\boldsymbol{v} - \tilde{\boldsymbol{v}}\|_2 + \|\mathbf{W} - \tilde{\mathbf{W}}\|_2$ *denotes the change of two set of parameters.*

Equipped with above intermediate results, we now proceed to prove Lemma 17.

Proof. Recall that our goal is to explore the impact of different GCN structures on the uniform stability constant ϵ_{APPNP}, which is a function of ρ_f, G_f, and L_f. By Lemma 19, we know that the function f is ρ_f-Lipschitz continuous, with $\rho_f = B_d^\alpha B_x$.

By Lemma 20, we know that the function f is L_f-smoothness, and the gradient of each weight matrices is bounded by G_f, with

$$G_f \leq \frac{4}{\gamma}B_d^\alpha B_x,\ L_f \leq \frac{4}{\gamma}B_d^\alpha B_x\Big(B_d^\alpha B_x \max\{1, B_w\} + 1\Big).$$

By plugging ϵ_{APPNP} into Theorem 3, we obtain the generalization bound of APPNP. □

H.1 Proof of Lemma 18

By the definition of $h_{\max}^{(L)}$, we have

$$\begin{aligned} h_{\max}^{(L)} &= \max_i \|[\alpha \mathbf{L}\mathbf{H}^{(L-1)} + (1-\alpha)\mathbf{H}^{(0)}]_{i,:}\|_2 \\ &\leq \alpha \max_i \|[\mathbf{L}\mathbf{H}^{(L-1)}]_{i,:}\|_2 + (1-\alpha)\max_i \|[\mathbf{H}^{(0)}]_{i,:}\|_2 \\ &= \alpha \max_i \left\| \sum_{j=1}^{N} L_{i,j}\mathbf{h}_j^{(L-1)} \right\|_2 + (1-\alpha)B_w B_x \\ &\underset{(a)}{\leq} \alpha\sqrt{d}h_{\max}^{(L-1)} + (1-\alpha)B_w B_x \\ &\underset{(b)}{\leq} \Big((1-\alpha)\sum_{\ell=1}^{L}(\alpha\sqrt{d})^{\ell-1} + (\alpha\sqrt{d})^L\Big)B_x B_w, \end{aligned} \tag{122}$$

where inequality (a) is follows from Lemma 31 and inequality (b) can be obtained by recursively applying (a).

Let denote $B_d^\alpha = (1-\alpha)\sum_{\ell=1}^{L}(\alpha\sqrt{d})^{\ell-1} + (\alpha\sqrt{d})^L$, then we have

$$h_{\max}^{(L)} \leq B_d^\alpha B_x B_w. \tag{123}$$

H.2 Proof of Lemma 19

By the definition of $\Delta h_{\max}^{(L)}$, we have

$$\begin{aligned} \Delta h_{\max}^{(L)} &= \max_i \|\mathbf{h}_i^{(L)} - \tilde{\mathbf{h}}_i^{(L)}\|_2 \\ &= \max_i \|[\big(\alpha\mathbf{L}\mathbf{H}^{(L-1)} + (1-\alpha)\mathbf{H}^{(0)}\big) - \big(\alpha\mathbf{L}\tilde{\mathbf{H}}^{(L-1)} + (1-\alpha)\tilde{\mathbf{H}}^{(0)}\big)]_{i,:}\|_2 \\ &\leq \alpha \max_i \|[\mathbf{L}(\mathbf{H}^{(L-1)} - \tilde{\mathbf{H}}^{(L-1)})]_{i,:}\|_2 + (1-\alpha)\max_i \|[\mathbf{H}^{(0)} - \tilde{\mathbf{H}}^{(0)}]_{i,:}\|_2 \\ &\underset{(a)}{\leq} \alpha\sqrt{d}\Delta h_{\max}^{(L-1)} + (1-\alpha)B_x\|\Delta\mathbf{W}\|_2 \\ &\underset{(b)}{\leq} \Big((1-\alpha)\sum_{L=1}^{L}(\alpha\sqrt{d})^{\ell-1} + (\alpha\sqrt{d})^{\ell}\Big)B_x\|\Delta\mathbf{W}\|_2, \end{aligned} \tag{124}$$

where inequality (a) is follows from Lemma 31 and inequality (b) can be obtained by recursively applying (a).

Let denote $B_d^\alpha = (1-\alpha)\sum_{\ell=1}^{L}(\alpha\sqrt{d})^{\ell-1} + (\alpha\sqrt{d})^L$, then we have

$$\Delta h_{\max}^{(L)} \leq B_d^\alpha B_x\|\Delta\mathbf{W}\|_2. \tag{125}$$

H.3 Proof of Lemma 20

By definition, we know that $\mathbf{H}^{(L)}$ is computed as

$$\mathbf{H}^{(L)} = \Big((\alpha\mathbf{L})^L + (1-\alpha)\big(1 + (\alpha\mathbf{L}) + \ldots + (\alpha\mathbf{L})^{L-1}\big)\Big)\mathbf{X}\mathbf{W}. \tag{126}$$

Therefore, the gradient with respect to weight $\mathbf{W}$ is computed as

$$\begin{aligned} \|\mathbf{G}^{(1)}\|_2 &\underset{(a)}{\leq} \frac{2}{\gamma}\max_i \left\|\frac{\partial f(\mathbf{h}_i)}{\partial \mathbf{W}}\right\|_2 \\ &= \frac{2}{\gamma}\max_i \left\|\Big[\Big((\alpha\mathbf{L})^L + (1-\alpha)\big(1 + (\alpha\mathbf{L}) + \ldots + (\alpha\mathbf{L})^{L-1}\big)\Big)\mathbf{X}\Big]_{i,:}\right\|_2 \\ &\leq \frac{2}{\gamma}\Big((\alpha\sqrt{d})^L + (1-\alpha)\big(1 + (\alpha\sqrt{d}) + \ldots + (\alpha\sqrt{d})^{L-1}\big)\Big)B_x \\ &\leq \frac{2}{\gamma}B_d^\alpha B_x, \end{aligned} \tag{127}$$

where inequality (a) is due to the fact that the loss function is $\frac{2}{\gamma}$-Lipschitz continuous, and $B_d^\alpha = (1-\alpha)\sum_{\ell=1}^{L}(\alpha\sqrt{d})^{\ell-1} + (\alpha\sqrt{d})^L$. Besides, the gradient with respect to weight $\boldsymbol{v}$ is computed as

$$\begin{aligned}\|\mathbf{G}^{(2)}\|_2 &= \left\|\frac{\partial \Phi_\gamma(-p(f(\mathbf{h}_i^{(L)}), y_i))}{\partial \boldsymbol{v}}\right\|_2 \\ &\leq \frac{2}{\gamma} h_{\max}^{(L)} \\ &\leq \frac{2}{\gamma} B_d^\alpha B_x B_w.\end{aligned} \tag{128}$$

Combine the result above, we have $\sum_{\ell=1}^{2}\|\mathbf{G}^{(\ell)}\|_2 \leq \frac{4}{\gamma} B_d^\alpha B_x B_w$. Then, the difference between two gradient computed on two different set of weight parameters is computed as

$$\begin{aligned}\|\mathbf{G}^{(1)} - \tilde{\mathbf{G}}^{(1)}\|_2 &\leq \frac{2}{\gamma}\max_i \left\|\frac{\partial f(\mathbf{h}_i^{(L)})}{\partial \mathbf{h}_i^{(L)}}\frac{\partial \mathbf{h}_i^{(L)}}{\partial \mathbf{W}} - \frac{\partial f(\tilde{\mathbf{h}}_i^{(L)})}{\partial \tilde{\mathbf{h}}_i^{(L)}}\frac{\partial \tilde{\mathbf{h}}_i^{(L)}}{\partial \tilde{\mathbf{W}}}\right\|_2 \\ &= \frac{2}{\gamma} B_x B_d^\alpha \left\|\frac{\partial f(\mathbf{h}_i^{(L)})}{\partial \mathbf{h}_i^{(L)}} - \frac{\partial f(\tilde{\mathbf{h}}_i^{(L)})}{\partial \tilde{\mathbf{h}}_i^{(L)}}\right\|_2 \\ &\underset{(a)}{\leq} \frac{2}{\gamma} B_x B_d^\alpha \Big(\Delta h_{\max}^{(L)} + (h_{\max}^{(L)}+1)\|\Delta \boldsymbol{v}\|_2\Big) \\ &\underset{(b)}{\leq} \frac{2}{\gamma} B_x B_d^\alpha \Big(B_x B_d^\alpha \|\Delta \mathbf{W}\|_2 + (B_x B_d^\alpha B_w + 1)\|\Delta \boldsymbol{v}\|_2\Big) \\ &\leq \frac{2}{\gamma} B_x B_d^\alpha \Big(B_x B_d^\alpha \max\{1, B_w\} + 1\Big)\|\Delta \boldsymbol{\theta}\|_2,\end{aligned} \tag{129}$$

where inequality (a) follows from Lemma 5 and inequality (b) follows from Lemma 19 and Lemma 18.

Similarly, the difference of gradient w.r.t. weight $\boldsymbol{v}$ is computed as

$$\begin{aligned}\|\mathbf{G}^{(2)} - \tilde{\mathbf{G}}^{(2)}\|_2 &= \left\|\frac{\partial \Phi_\gamma(-p(f(\mathbf{h}_i^{(L)}), y_i))}{\partial \boldsymbol{v}} - \frac{\partial \Phi_\gamma(-p(\tilde{f}(\tilde{\mathbf{h}}_i^{(L)}), y_i))}{\partial \tilde{\boldsymbol{v}}}\right\|_2 \\ &\leq \frac{2}{\gamma}\Delta h_{\max}^{(L)} \\ &\leq \frac{2}{\gamma} B_d^\alpha B_x \|\Delta \mathbf{W}\|_2.\end{aligned} \tag{130}$$

Combining result above yields $\sum_{\ell=1}^{2}\|\mathbf{G}^{(\ell)} - \tilde{\mathbf{G}}^{(\ell)}\|_2 \leq \frac{4}{\gamma} B_x B_d^\alpha \Big(B_x B_d^\alpha \max\{1, B_w\} + 1\Big)\|\Delta \boldsymbol{\theta}\|_2$ as desired.

I Generalization bound for GCNII

In the following, we provide generalization bound of the GCNII. Recall that the update rule of GCNII is defined as

$$\mathbf{H}^{(\ell)} = \sigma\Big(\big(\alpha \mathbf{L}\mathbf{H}^{(\ell)} + (1-\alpha)\mathbf{X}\big)\big(\beta \mathbf{W}^{(\ell)} + (1-\beta)\mathbf{I}_N\big)\Big), \tag{131}$$

where $\sigma(\cdot)$ is ReLU activation function. Although ReLU function $\sigma(x)$ is not differentiable when $x = 0$, for analysis purpose, we suppose the $\sigma'(0) = 0$ which is also used in optimization.

The training of GCNII is an empirical risk minimization with respect to a set of parameters $\boldsymbol{\theta} = \{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L)}, \boldsymbol{v}\}$, i.e.,

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m}\Phi_\gamma(-p(f(\mathbf{h}_i^{(L)}), y_i)),\ f(\mathbf{h}_i^{(L)}) = \tilde{\sigma}(\boldsymbol{v}^\top \mathbf{h}_i^{(L)}), \tag{132}$$

where $\mathbf{h}_i^{(L)}$ is the node representation of the ith node at the final layer, $f(\mathbf{h}_i^{(L)})$ is the predicted label for the ith node, $\tilde{\sigma}(x) = \frac{1}{\exp(-x)+1}$ is the sigmoid function, and loss function $\Phi_\gamma(-p(z,y))$ is $\frac{2}{\gamma}$-Lipschitz continuous with respect to its first input z. For simplification, let $\text{Loss}(z,y)$ denote $\Phi_\gamma(-p(z,y))$ in the proof.

To establish the generalization of GCNII as stated in Theorem 3, we utilize the result on transductive uniform stability from [16] (Theorem 8 in Appendix F). Then, in Lemma 21, we derive the uniform stability constant for GCNII, i.e., ϵ_{GCNII}.

Lemma 21. *The uniform stability constant for GCNII is computed as* $\epsilon_{\texttt{GCNII}} = \frac{2\eta\rho_f G_f}{m}\sum_{t=1}^{T}(1+\eta L_F)^{t-1}$ *where*

$$\begin{aligned}
&\rho_f = \beta C_1^L C_2,\ G_f = \beta\frac{2}{\gamma}(L+1)C_1^L C_2,\\
&L_f = \alpha\beta\frac{2}{\gamma}(L+1)\sqrt{d}C_1^L C_2\Big((\alpha\beta L+\beta B_w+2)C_1^L C_2+2\beta\Big),\\
&C_1 = \max\{1,\alpha\sqrt{d}B_w^\beta\},\ C_2 = \sqrt{d}+B_{\ell,d}^{\alpha,\beta}B_x,\\
&B_w^\beta = \beta B_w+(1-\beta),\ B_{\ell,d}^{\alpha,\beta} = \max\Big\{\beta\big((1-\alpha)L+\alpha\sqrt{d}\big),(1-\alpha)LB_w^\beta+1\Big\}.
\end{aligned} \tag{133}$$

By plugging the result in Lemma 21 back to Theorem 8, we establish the generalization bound for GCNII.

The key idea of the proof is to decompose the change of the GCNII output into two terms (in Lemma 5) which depend on

- (Lemma 22) The maximum change of node representation $\Delta h_{\max}^{(\ell)} = \max_i \|[\mathbf{H}^{(\ell)} - \tilde{\mathbf{H}}^{(\ell)}]_{i,:}\|_2$,
- (Lemma 23) The maximum node representation $h_{\max}^{(\ell)} = \max_i \|[\mathbf{H}^{(\ell)}]_{i,:}\|_2$.

Lemma 22 (Upper bound of $h_{\max}^{(\ell)}$ for **GCNII**)**.** *Let suppose Assumption 1 hold. Then, the maximum node embeddings for any node at the ℓth layer is bounded by*

$$h_{\max}^{(\ell)} \le \big((1-\alpha)\ell B_w^\beta+1\big)B_x\cdot\big(\max\{1,\alpha\sqrt{d}B_w^\beta\}\big)^\ell, \tag{134}$$

where $B_w^\beta = \beta B_w+(1-\beta)$ *and* $\Delta\mathbf{W}^{(\ell)} = \mathbf{W}^{(\ell)}-\tilde{\mathbf{W}}^{(\ell)}$.

Lemma 23 (Upper bound of $\Delta h_{\max}^{(\ell)}$ for **GCNII**)**.** *Let suppose Assumption 1 hold. Then, the maximum change between the node embeddings on two different set of weight parameters for any node at the ℓth layer is bounded by*

$$\Delta h_{\max}^{(\ell)} \le \beta B_x\big((1-\alpha)\ell+\alpha\sqrt{d}\big)\big(\max\{1,\alpha\sqrt{d}B_w^\beta\}\big)^{\ell-1}\Big(\|\Delta\mathbf{W}^{(1)}\|_2+\ldots+\|\Delta\mathbf{W}^{(\ell)}\|_2\Big), \tag{135}$$

where $B_w^\beta = \beta B_w+(1-\beta)$ *and* $\Delta\mathbf{W}^{(\ell)} = \mathbf{W}^{(\ell)}-\tilde{\mathbf{W}}^{(\ell)}$.

Besides, in Lemma 24, we derive the upper bound on the maximum change of node embeddings before the activation function, i.e., $\Delta z_{\max}^{(\ell)} = \max_i \|[\mathbf{Z}^{(\ell)} - \tilde{\mathbf{Z}}^{(\ell)}]_{i,:}\|_2$, which will be used to derive the gradient related upper bounds.

Lemma 24 (Upper bound of $\Delta z_{\max}^{(\ell)}$ for **GCNII**)**.** *Let suppose Assumption 1 hold. Then, the maximum change between the node embeddings before the activation function on two different set of weight parameters for any node at the ℓth layer is bounded by*

$$\Delta z_{\max}^{(\ell)} \le \beta B_x\big((1-\alpha)\ell+\alpha\sqrt{d}\big)(\max\{1,\alpha\sqrt{d}B_w^\beta\})^{\ell-1}\Big(\|\Delta\mathbf{W}^{(1)}\|_2+\ldots+\|\Delta\mathbf{W}^{(\ell)}\|_2\Big), \tag{136}$$

where $B_w^\beta = \beta B_w+(1-\beta)$ *and* $\Delta\mathbf{W}^{(\ell)} = \mathbf{W}^{(\ell)}-\tilde{\mathbf{W}}^{(\ell)}$.

Then, in Lemma 25, we decompose the change of the model parameters into two terms which depend on

- The maximum change of gradient passing from $(\ell+1)$th layer to the ℓth layer $\Delta d_{\max}^{(\ell)} = \max_i \left\| \left[\frac{\partial \mathbf{H}^{(\ell)}}{\partial \mathbf{H}^{(\ell-1)}} - \frac{\partial \tilde{\mathbf{H}}^{(\ell)}}{\partial \tilde{\mathbf{H}}^{(\ell-1)}} \right]_{i,:} \right\|_2$,
- The maximum gradient passing from $(\ell+1)$th layer to the ℓth layer $d_{\max}^{(\ell)} = \max_i \left\| \left[\frac{\partial \mathbf{H}^{(\ell)}}{\partial \mathbf{H}^{(\ell-1)}} \right]_{i,:} \right\|_2$.

Lemma 25 (Upper bound of $\mathbf{d}_{\max}^{(\ell)}, \Delta\mathbf{d}_{\max}^{(\ell)}$ for **GCNII**). *Let suppose Assumption 1 hold and let denote* $C_1 = \max\{1, \alpha\sqrt{d}B_w^\beta\}$, $C_2 = \sqrt{d} + B_{\ell,d}^{\alpha,\beta} B_x$, $B_w^\beta = \beta B_w + (1-\beta)$, *and* $B_{\ell,d}^{\alpha,\beta} = \max\left\{\beta\big((1-\alpha)L + \alpha\sqrt{d}\big), (1-\alpha)LB_w^\beta + 1\right\}$. *Then, the maximum gradient passing from layer* $\ell+1$ *to layer* ℓ *for any node is bounded by*

$$d_{\max}^{(\ell)} \leq \frac{2}{\gamma} C_1^{L-\ell+1} \|\Delta\boldsymbol{\theta}\|_2, \tag{137}$$

and the maximum change between the gradient passing from layer $\ell+1$ *to layer* ℓ *on two different set of weight parameters for any node is bounded by*

$$\Delta d_{\max}^{(\ell)} \leq \frac{2}{\gamma} C_1^{L-\ell}\Big((\alpha\beta L + \beta B_w + 1)C_1^L C_2 + 1\Big)\|\Delta\boldsymbol{\theta}\|_2, \tag{138}$$

where $\|\Delta\boldsymbol{\theta}\|_2 = \|\boldsymbol{v} - \tilde{\boldsymbol{v}}\|_2 + \sum_{\ell-1}^L \|\mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)}\|_2$ *denotes the change of two set of parameters.*

Finally, in Lemma 26, we decompose the change of the model parameters into two terms which depend on

- The change of gradient with respect to the ℓth layer weight matrix $\|\Delta\mathbf{G}^{(\ell)}\|_2$,
- The gradient with respect to the ℓth layer weight matrix $\|\mathbf{G}^{(\ell)}\|_2$,

where $\mathbf{G}^{(L+1)}$ denotes the gradient with respect to the weight $\boldsymbol{v}$ of the binary classifier and $\mathbf{G}^{(\ell)}$ denotes the gradient with respect to the weight matrices $\mathbf{W}^{(\ell)}$ of the graph convolutional layer. Notice that $\|\Delta\mathbf{G}^{(\ell)}\|_2$ reflect the smoothness of GCN model and $\|\mathbf{G}^{(\ell)}\|_2$ correspond the upper bound of gradient.

Lemma 26 (Upper bound of $\mathbf{G}^{(\ell)}, \Delta\mathbf{G}^{(\ell)}$ for **GCNII**). *Let suppose Assumption 1 hold, and let* $C_1 = \max\{1, \alpha\sqrt{d}B_w^\beta\}$, $C_2 = \sqrt{d} + B_{\ell,d}^{\alpha,\beta} B_x$, $B_w^\beta = \beta B_w + (1-\beta)$, *and* $B_{\ell,d}^{\alpha,\beta} = \max\left\{\beta\big((1-\alpha)L + \alpha\sqrt{d}\big), (1-\alpha)LB_w^\beta + 1\right\}$. *Then, the gradient and the maximum change between gradients on two different set of weight parameters are bounded by*

$$\begin{aligned}
&\sum_{\ell=1}^{L+1} \|\mathbf{G}^{(\ell)}\|_2 \leq \frac{2}{\gamma}(L+1)\beta C_1^L C_2, \\
&\sum_{\ell=1}^{L+1} \|\Delta\mathbf{G}^{(\ell)}\|_2 \leq \alpha\beta\frac{2}{\gamma}(L+1)\sqrt{d}C_1^L C_2\Big((\alpha\beta L + \beta B_w + 2)C_1^L C_2 + 2\beta\Big)\|\Delta\boldsymbol{\theta}\|_2,
\end{aligned} \tag{139}$$

where $\|\Delta\boldsymbol{\theta}\|_2 = \|\boldsymbol{v} - \tilde{\boldsymbol{v}}\|_2 + \sum_{\ell-1}^L \|\mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)}\|_2$ *denotes the change of two set of parameters.*

Equipped with above intermediate results, we now proceed to prove Lemma 21.

Proof. Recall that our goal is to explore the impact of different GCN structures on the uniform stability constant ϵ_{GCNII}, which is a function of ρ_f, G_f, and L_f. Let denote $C_1 = \max\{1, \alpha\sqrt{d}B_w^\beta\}$, $C_2 = \sqrt{d} + B_{\ell,d}^{\alpha,\beta} B_x$, $B_w^\beta = \beta B_w + (1-\beta)$, and $B_{\ell,d}^{\alpha,\beta} = \max\left\{\beta\big((1-\alpha)L + \alpha\sqrt{d}\big), (1-\alpha)LB_w^\beta + 1\right\}$. By Lemma 23, we know that the function f is ρ_f-Lipschitz continuous, with

$$\rho_f = \beta C_1^L C_2. \tag{140}$$

By Lemma 26, we know that the function f is L_f-smoothness, and the gradient of each weight matrices is bounded by G_f, with

$$\begin{aligned}
G_f &= \frac{2}{\gamma}(L+1)\beta C_1^L C_2, \\
L_f &= \alpha\beta\frac{2}{\gamma}(L+1)\sqrt{d}C_1^L C_2\Big((\alpha\beta L + \beta B_w + 2)C_1^L C_2 + 2\beta\Big).
\end{aligned} \tag{141}$$

where $B_{\ell,d}^{\alpha,\beta} = \max\left\{\beta\big((1-\alpha)L + \alpha\sqrt{d}\big), (1-\alpha)LB_w^\beta + 1\right\}$.

By plugging ϵ_{GCNII} into Theorem 3, we obtain the generalization bound of GCNII.

□

I.1 Proof of Lemma 22

By the definition of $h_{\max}^{(\ell)}$, we have

$$\begin{aligned}
h_{\max}^{(\ell)} &= \max_i \|[\sigma\Big(\big(\alpha\mathbf{L}\mathbf{H}^{(\ell-1)} + (1-\alpha)\mathbf{X}\big)\big(\beta\mathbf{W}^{(\ell)} + (1-\beta)\mathbf{I}\big)\Big)]_{i,:}\|_2 \\
&\underset{(a)}{\leq} \max_i \|[\big(\alpha\mathbf{L}\mathbf{H}^{(\ell-1)} + (1-\alpha)\mathbf{X}\big)\big(\beta\mathbf{W}^{(\ell)} + (1-\beta)\mathbf{I}\big)]_{i,:}\|_2 \\
&\leq \max_i \|[\big(\alpha\mathbf{L}\mathbf{H}^{(\ell-1)} + (1-\alpha)\mathbf{X}\big)]_{i,:}\|_2 \|\beta\mathbf{W}^{(\ell)} + (1-\beta)\mathbf{I}\|_2 \\
&\leq (\beta B_w + (1-\beta)) \cdot \max_i \|[\big(\alpha\mathbf{L}\mathbf{H}^{(\ell-1)} + (1-\alpha)\mathbf{X}\big)]_{i,:}\|_2 \\
&\leq (\beta B_w + (1-\beta)) \cdot \Big(\alpha \max_i \|[\mathbf{L}\mathbf{H}^{(\ell-1)}]_{i,:}\|_2 + (1-\alpha)\max_i \|[\mathbf{X}]_{i,:}\|_2\Big) \\
&\leq (\beta B_w + (1-\beta)) \cdot \Big(\alpha\sqrt{d}h_{\max}^{(\ell-1)} + (1-\alpha)B_x\Big) \\
&= \alpha\sqrt{d}(\beta B_w + (1-\beta))h_{\max}^{(\ell-1)} + (1-\alpha)B_x(\beta B_w + (1-\beta)),
\end{aligned} \tag{142}$$

where inequality (a) is due to $\|\sigma(x)\|_2 \leq \|x\|_2$.

Let $B_w^\beta = \beta B_w + (1-\beta)$. By induction, we have

$$h_{\max}^{(\ell)} \leq \Big((1-\alpha)B_x B_w^\beta\Big)\sum_{\ell'=1}^{\ell}\Big(\alpha\sqrt{d}B_w^\beta\Big)^{\ell'-1} + \Big(\alpha\sqrt{d}B_w^\beta\Big)^{\ell} B_x. \tag{143}$$

If $\alpha\sqrt{d}B_w^\beta \leq 1$ we have,

$$h_{\max}^{(\ell)} \leq (1-\alpha)B_x B_w^\beta \ell + B_x = \Big((1-\alpha)\ell B_w^\beta + 1\Big)B_x, \tag{144}$$

and if $\alpha\sqrt{d}B_w^\beta > 1$ we have

$$\begin{aligned}
h_{\max}^{(\ell)} &\leq \Big((1-\alpha)B_x B_w^\beta\Big)\ell\Big(\alpha\sqrt{d}B_w^\beta\Big)^{\ell-1} + \Big(\alpha\sqrt{d}B_w^\beta\Big)^{\ell} B_x \\
&\leq \Big((1-\alpha)\ell B_w^\beta + 1\Big)B_x \cdot \Big(\alpha\sqrt{d}B_w^\beta\Big)^{\ell}.
\end{aligned} \tag{145}$$

By combining results above, we have

$$h_{\max}^{(\ell)} \leq \Big((1-\alpha)\ell B_w^\beta + 1\Big)B_x \cdot \Big(\max\{1, \alpha\sqrt{d}B_w^\beta\}\Big)^{\ell}. \tag{146}$$

I.2 Proof of Lemma 23

By the definition of $\Delta h_{\max}^{(\ell)}$, we have

$$
\begin{aligned}
\Delta h_{\max}^{(\ell)} &= \max_i \|\mathbf{h}_i^{(\ell)} - \tilde{\mathbf{h}}_i^{(\ell)}\|_2 \\
&= \max_i \Big\| \Big[\sigma\Big(\big(\alpha \mathbf{L}\mathbf{H}^{(\ell-1)} + (1-\alpha)\mathbf{X}\big)\big(\beta \mathbf{W}^{(\ell)} + (1-\beta)\mathbf{I}\big)\Big) \\
&\quad - \sigma\Big(\big(\alpha \mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)} + (1-\alpha)\mathbf{X}\big)\big(\beta \tilde{\mathbf{W}}^{(\ell)} + (1-\beta)\mathbf{I}\big)\Big)\Big]_{i,:} \Big\|_2 \\
&\le \max_i \Big\| \Big[\big(\alpha \mathbf{L}\mathbf{H}^{(\ell-1)} + (1-\alpha)\mathbf{X}\big)\big(\beta \mathbf{W}^{(\ell)} + (1-\beta)\mathbf{I}\big) \\
&\quad - \big(\alpha \mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)} + (1-\alpha)\mathbf{X}\big)\big(\beta \tilde{\mathbf{W}}^{(\ell)} + (1-\beta)\mathbf{I}\big)\Big]_{i,:} \Big\|_2 \\
&\underset{(a)}{\le} \max_i \Big\| \Big[\big(\alpha \mathbf{L}\mathbf{H}^{(\ell-1)} + (1-\alpha)\mathbf{X}\big)\big(\beta(\mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)})\big)\Big]_{i,:} \Big\|_2 \\
&\quad + \max_i \Big\| \Big[\big(\alpha \mathbf{L}(\mathbf{H}^{(\ell-1)} - \tilde{\mathbf{H}}^{(\ell-1)})\big)\big(\beta \tilde{\mathbf{W}}^{(\ell)} + (1-\beta)\mathbf{I}\big)\Big]_{i,:} \Big\|_2 \\
&\le \max_i \|[\alpha \mathbf{L}\mathbf{H}^{(\ell-1)} + (1-\alpha)\mathbf{X}]_{i,:}\|_2 \cdot \beta \|\Delta \mathbf{W}^{(\ell)}\|_2 + \alpha(\beta B_w + (1-\beta)) \max_i \|[\mathbf{L}(\mathbf{H}^{(\ell-1)} - \tilde{\mathbf{H}}^{(\ell-1)})]_{i,:}\|_2 \\
&\le \Big(\alpha\sqrt{d} h_{\max}^{(\ell-1)} + (1-\alpha)B_x\Big) \cdot \beta \|\Delta \mathbf{W}^{(\ell)}\|_2 + \alpha\big(\beta B_w + (1-\beta)\big)\sqrt{d}\Delta h_{\max}^{(\ell-1)},
\end{aligned}
\tag{147}
$$

where (a) is due to the fact that $\|\sigma(x)\|_2 \le \|x\|_2$.

Let $B_w^\beta = \beta B_w + (1-\beta)$. Then, by induction we have

$$
\begin{aligned}
\Delta h_{\max}^{(\ell)} &\le \alpha\sqrt{d} B_w^\beta \cdot \Delta h_{\max}^{(\ell-1)} + \beta\Big(\alpha\sqrt{d} h_{\max}^{(\ell-1)} + (1-\alpha)B_x\Big)\|\Delta \mathbf{W}^{(\ell)}\|_2 \\
&\le (\alpha\sqrt{d} B_w^\beta)^2 \cdot \Delta h_{\max}^{(\ell-2)} + \beta\Big(\alpha\sqrt{d} h_{\max}^{(\ell-1)} + (1-\alpha)B_x\Big)\|\Delta \mathbf{W}^{(\ell)}\|_2 \\
&\quad + \beta(\alpha\sqrt{d} B_w^\beta)\Big(\alpha\sqrt{d} h_{\max}^{(\ell-2)} + (1-\alpha)B_x\Big)\|\Delta \mathbf{W}^{(\ell-1)}\|_2 \\
&\cdots \\
&\le \beta\Big(\alpha\sqrt{d} h_{\max}^{(\ell-1)} + (1-\alpha)B_x\Big)\|\Delta \mathbf{W}^{(\ell)}\|_2 \\
&\quad + \beta(\alpha\sqrt{d} B_w^\beta)\Big(\alpha\sqrt{d} h_{\max}^{(\ell-2)} + (1-\alpha)B_x\Big)\|\Delta \mathbf{W}^{(\ell-1)}\|_2 + \ldots \\
&\quad + \beta(\alpha\sqrt{d} B_w^\beta)^{\ell-1}\Big(\alpha\sqrt{d} h_{\max}^{(0)} + (1-\alpha)B_x\Big)\|\Delta \mathbf{W}^{(1)}\|_2.
\end{aligned}
\tag{148}
$$

If $\alpha\sqrt{d} B_w^\beta \le 1$, using Lemma 23 we have

$$
\begin{aligned}
\Delta h_{\max}^{(\ell)} &\le \beta\Big(\alpha\sqrt{d} h_{\max}^{(\ell-1)} + (1-\alpha)B_x\Big)\|\Delta \mathbf{W}^{(\ell)}\|_2 + \beta\Big(\alpha\sqrt{d} h_{\max}^{(\ell-2)} + (1-\alpha)B_x\Big)\|\Delta \mathbf{W}^{(\ell-1)}\|_2 + \ldots \\
&\quad + \beta\Big(\alpha\sqrt{d} h_{\max}^{(0)} + (1-\alpha)B_x\Big)\|\Delta \mathbf{W}^{(1)}\|_2 \\
&\le \beta B_x\Big((1-\alpha)\ell + \alpha\sqrt{d}\Big)\Big(\|\Delta \mathbf{W}^{(1)}\|_2 + \ldots + \|\Delta \mathbf{W}^{(\ell)}\|_2\Big),
\end{aligned}
\tag{149}
$$

and if $\alpha\sqrt{d} B_w^\beta > 1$, using Lemma 23 we have

$$
\begin{aligned}
\Delta h_{\max}^{(\ell)} &\le \beta\Big(\alpha\sqrt{d}\big((1-\alpha)(\ell-1)B_w^\beta + 1\big)B_x \cdot \big(\alpha\sqrt{d} B_w^\beta\big)^{\ell-1} + (1-\alpha)B_x\Big)\|\Delta \mathbf{W}^{(\ell)}\|_2 \\
&\quad + \beta(\alpha\sqrt{d} B_w^\beta)\Big(\alpha\sqrt{d}\big((1-\alpha)(\ell-2)B_w^\beta + 1\big)B_x \cdot \big(\alpha\sqrt{d} B_w^\beta\big)^{\ell-2} + (1-\alpha)B_x\Big)\|\Delta \mathbf{W}^{(\ell-1)}\|_2 + \ldots \\
&\quad + \beta(\alpha\sqrt{d} B_w^\beta)^{\ell-1}\Big(\alpha\sqrt{d} B_x + (1-\alpha)B_x\Big)\|\Delta \mathbf{W}^{(1)}\|_2 \\
&\le \beta B_x\big((1-\alpha)\ell + \alpha\sqrt{d}\big)(\alpha\sqrt{d} B_w^\beta)^{\ell-1}\Big(\|\Delta \mathbf{W}^{(1)}\|_2 + \ldots + \|\Delta \mathbf{W}^{(\ell)}\|_2\Big).
\end{aligned}
\tag{150}
$$

By combining the result above, we have

$$\Delta h_{\max}^{(\ell)} \leq \beta B_x \big((1-\alpha)\ell + \alpha\sqrt{d}\big)(\max\{1, \alpha\sqrt{d}B_w^\beta\})^{\ell-1}\Big(\|\Delta\mathbf{W}^{(1)}\|_2 + \ldots + \|\Delta\mathbf{W}^{(\ell)}\|_2\Big). \quad (151)$$

I.3 Proof of Lemma 24

The proof is similar to the proof of Lemma 23.

I.4 Proof of Lemma 25

Let $B_w^\beta = \beta B_w + (1-\beta)$. By the definition of $d_{\max}^{(\ell)}$, we have

$$\begin{aligned}
d_{\max}^{(\ell)} &= \max_i \|[\mathbf{D}^{(\ell)}]_{i,:}\|_2 \\
&= \alpha \max_i \|[\mathbf{L}^\top \sigma'(\mathbf{Z}^{(\ell)}) \odot \mathbf{D}^{(\ell+1)}\big(\beta\mathbf{W}^{(\ell)} + (1-\beta)\mathbf{I}\big)]_{i,:}\|_2 \\
&\leq \alpha \max_i \|[\mathbf{L}^\top \sigma'(\mathbf{Z}^{(\ell)}) \odot \mathbf{D}^{(\ell+1)}]_{i,:}\|_2 \cdot B_w^\beta \\
&\underset{(a)}{\leq} \alpha\sqrt{d}B_w^\beta \cdot d_{\max}^{(\ell+1)} \\
&\leq \frac{2}{\gamma}(\max\{1, \alpha\sqrt{d}B_w^\beta\})^{L-\ell+1},
\end{aligned} \quad (152)$$

where inequality (a) is due to the gradient of ReLU activation is element-wise either 0 and 1.

By the definition of $\Delta d_{\max}^{(\ell)}$, we have

$$\begin{aligned}
\Delta d_{\max}^{(\ell)} &= \max_i \|[\mathbf{D}^{(\ell)} - \tilde{\mathbf{D}}^{(\ell)}]_{i,:}\|_2 \\
&\leq \max_i \alpha \|[\mathbf{L}^\top \sigma'(\mathbf{Z}^{(\ell)}) \odot \mathbf{D}^{(\ell+1)}\big(\beta\mathbf{W}^{(\ell)} + (1-\beta)\mathbf{I}\big) - \mathbf{L}^\top \sigma'(\tilde{\mathbf{Z}}^{(\ell)}) \odot \tilde{\mathbf{D}}^{(\ell+1)}\big(\beta\tilde{\mathbf{W}}^{(\ell)} - (1-\beta)\mathbf{I}\big)]_{i,:}\|_2 \\
&\underset{(a)}{\leq} \max_i \alpha \|[\mathbf{L}^\top \mathbf{D}^{(\ell+1)}\big(\beta\mathbf{W}^{(\ell)} + (1-\beta)\mathbf{I}\big) - \mathbf{L}^\top \tilde{\mathbf{D}}^{(\ell+1)}\big(\beta\mathbf{W}^{(\ell)} + (1-\beta)\mathbf{I}\big)]_{i,:}\|_2 \\
&\quad + \max_i \alpha \|[\mathbf{L}^\top \tilde{\mathbf{D}}^{(\ell+1)}\big(\beta\mathbf{W}^{(\ell)} + (1-\beta)\mathbf{I}\big)]_{i,:} - \mathbf{L}^\top \tilde{\mathbf{D}}^{(\ell+1)}\big(\beta\tilde{\mathbf{W}}^{(\ell)} - (1-\beta)\mathbf{I}\big)]_{i,:}\|_2 \\
&\quad + \alpha \max_i \|[\mathbf{L}^\top \tilde{\mathbf{D}}^{(\ell+1)}\big(\beta\tilde{\mathbf{W}}^{(\ell)} - (1-\beta)\mathbf{I}\big)]_{i,:}\|_2 \max_i \|[\mathbf{Z}^{(\ell)} - \tilde{\mathbf{Z}}^{(\ell)}]_{i,:}\|_2 \\
&\underset{(b)}{\leq} \max_i \alpha \|[\mathbf{L}^\top (\mathbf{D}^{(\ell+1)} - \tilde{\mathbf{D}}^{(\ell+1)})\big(\beta\mathbf{W}^{(\ell)} + (1-\beta)\mathbf{I}\big)]_{i,:}\|_2 \\
&\quad + \max_i \alpha \|[\mathbf{L}^\top \tilde{\mathbf{D}}^{(\ell+1)}\big(\beta(\mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)})\big)]_{i,:}\|_2 \\
&\quad + \alpha \max_i \|[\mathbf{L}^\top \tilde{\mathbf{D}}^{(\ell+1)}\big(\beta\tilde{\mathbf{W}}^{(\ell)} - (1-\beta)\mathbf{I}\big)]_{i,:}\|_2 \max_i \|[\mathbf{Z}^{(\ell)} - \tilde{\mathbf{Z}}^{(\ell)}]_{i,:}\|_2 \\
&\leq \alpha\sqrt{d}B_w^\beta \cdot \Delta d_{\max}^{(\ell+1)} + \alpha\beta\sqrt{d}d_{\max}^{(\ell+1)}\|\Delta\mathbf{W}^{(\ell)}\|_2 + \alpha\sqrt{d}B_w^\beta d_{\max}^{(\ell+1)}\Delta z_{\max}^{(\ell)} \\
&= \alpha\sqrt{d}B_w^\beta \cdot \Delta d_{\max}^{(\ell+1)} + \underbrace{\alpha d_{\max}^{(\ell+1)}\Big(\beta\sqrt{d}\|\Delta\mathbf{W}^{(\ell)}\|_2 + \sqrt{d}B_w^\beta \Delta z_{\max}^{(\ell)}\Big)}_{(A)},
\end{aligned} \quad (153)$$

where inequality (a) and (b) is due to the gradient of ReLU activation is element-wise either 0 and 1.

Let $C_1 = \max\{1, \alpha\sqrt{d}B_w^\beta\}$ and $B_{\ell,d}^{\alpha,\beta} = \max\big\{\beta\big((1-\alpha)L + \alpha\sqrt{d}\big), (1-\alpha)LB_w^\beta + 1\big\}$. Recall that $d_{\max}^{(\ell+1)} \leq \frac{2}{\gamma}(\max\{1, \alpha\sqrt{d}B_w^\beta\})^{L-\ell}$ and the upper bound of $\Delta z_{\max}^{(\ell)}$ in Lemma 24, we have

$$d_{\max}^{(\ell+1)} \leq \frac{2}{\gamma}C_1^{L-\ell}, \ \Delta z_{\max}^{(\ell)} \leq \beta B_x B_{\ell,d}^{\alpha,\beta} C_1^{\ell-1}\big(\|\Delta\mathbf{W}^{(1)}\|_2 + \ldots + \|\Delta\mathbf{W}^{(\ell)}\|_2\big). \quad (154)$$

Therefore, we have upper bound (A) by

$$\begin{aligned}
&\alpha\sqrt{d}d_{\max}^{(\ell+1)}\Big(\beta\|\Delta\mathbf{W}^{(\ell)}\|_2 + B_w^{\beta}\Delta z_{\max}^{(\ell)}\Big) \\
&\leq \alpha\beta\cdot\frac{2}{\gamma}C_1^{L-\ell}\cdot\Big(\sqrt{d}\|\Delta\mathbf{W}^{(\ell)}\|_2 + B_x B_{\ell,d}^{\alpha,\beta}C_1^{\ell}\big(\|\Delta\mathbf{W}^{(1)}\|_2 + \ldots + \|\Delta\mathbf{W}^{(\ell)}\|_2\big)\Big) \\
&\leq \alpha\beta\frac{2}{\gamma}C_1^{L}\big(\sqrt{d} + B_{\ell,d}^{\alpha,\beta}B_x\big)\|\Delta\boldsymbol{\theta}\|_2 \\
&\leq \alpha\beta\frac{2}{\gamma}C_1^{L}C_2\|\Delta\boldsymbol{\theta}\|_2,
\end{aligned} \tag{155}$$

where $C_2 = \sqrt{d} + B_{\ell,d}^{\alpha,\beta}B_x$ and $B_{\ell,d}^{\alpha,\beta} = \max\Big\{\beta\big((1-\alpha)L + \alpha\sqrt{d}\big), (1-\alpha)LB_w^{\beta} + 1\Big\}$. By plugging it back and by induction, we have

$$\begin{aligned}
\Delta d_{\max}^{(\ell)} &\leq C_1\Delta d_{\max}^{(\ell+1)} + \alpha\beta\frac{2}{\gamma}C_1^{L}C_2\|\Delta\boldsymbol{\theta}\|_2 \\
&\leq C_1^2\Delta d_{\max}^{(\ell+2)} + (1+C_1)\cdot\alpha\beta\frac{2}{\gamma}C_1^{L}C_2\|\Delta\boldsymbol{\theta}\|_2 \\
&\leq C_1^{L-\ell+1}\Delta d_{\max}^{(L+1)} + (1 + C_1 + \ldots + C_1^{L-\ell+1})\cdot\alpha\beta\frac{2}{\gamma}C_1^{L}C_2\|\Delta\boldsymbol{\theta}\|_2 \\
&\leq C_1^{L-\ell}\Big(\Delta d_{\max}^{(L+1)} + \alpha\beta\frac{2}{\gamma}LC_1^{L}C_2\|\Delta\boldsymbol{\theta}\|_2\Big).
\end{aligned} \tag{156}$$

Let first explicit upper bound $\Delta d_{\max}^{(L+1)}$. By definition, $\Delta d_{\max}^{(L+1)}$ as

$$\begin{aligned}
\Delta d_{\max}^{(L+1)} &= \max_i \|\mathbf{d}_i^{(L+1)} - \tilde{\mathbf{d}}_i^{(L+1)}\|_2 \\
&= \max_i \left\|\frac{\partial \text{Loss}(f(\mathbf{h}_i^{(L)}), y_i)}{\partial \mathbf{h}_i^{(L)}} - \frac{\partial \text{Loss}(\tilde{f}(\tilde{\mathbf{h}}_i^{(L)}), y_i)}{\partial \tilde{\mathbf{h}}_i^{(L)}}\right\|_2 \\
&\leq \frac{2}{\gamma}\max_i \left\|\frac{\partial f(\mathbf{h}_i^{(L)})}{\partial \mathbf{h}_i^{(L)}} - \frac{\partial f(\tilde{\mathbf{h}}_i^{(L)})}{\partial \tilde{\mathbf{h}}_i^{(L)}}\right\|_2 \\
&\leq \frac{2}{\gamma}\Big(\Delta h_{\max}^{(L)} + (h_{\max}^{(L)} + 1)\Delta\boldsymbol{v}\Big) \\
&= \frac{2}{\gamma}\Big[B_x B_{\ell,d}^{\alpha,\beta}C_1^{L-1}\big(\|\Delta\mathbf{W}^{(1)}\|_2 + \ldots + \|\Delta\mathbf{W}^{(L)}\|_2\big) + \big(B_x B_{\ell,d}^{\alpha,\beta}C_1^{L} + 1\big)\|\Delta\boldsymbol{v}\|_2\Big] \\
&\leq \frac{2}{\gamma}\Big(B_\ell^{\alpha}B_x C_1^{L} + 1\Big)\|\Delta\boldsymbol{\theta}\|_2 \\
&= \frac{2}{\gamma}\Big(B_\ell^{\alpha}B_x C_1^{L} + 1\Big)\|\Delta\boldsymbol{\theta}\|_2,
\end{aligned} \tag{157}$$

where $C_1 = \max\{1, \alpha\sqrt{d}B_w^{\beta}\}$, $B_w^{\beta} = \beta B_w + (1-\beta)$ and $B_{\ell,d}^{\alpha,\beta} = \max\Big\{\beta\big((1-\alpha)L + \alpha\sqrt{d}\big), (1-\alpha)LB_w^{\beta} + 1\Big\}$. By plugging it back, we have

$$\begin{aligned}
\Delta d_{\max}^{(\ell)} &\leq C_1^{L-\ell}\Big(\frac{2}{\gamma}\big(B_{\ell,d}^{\alpha,\beta}B_x C_1^{L} + 1\big)\|\Delta\boldsymbol{\theta}\|_2 + \alpha\beta\frac{2}{\gamma}LC_1^{L}(\sqrt{d} + B_{\ell,d}^{\alpha,\beta}B_x)\|\Delta\boldsymbol{\theta}\|_2\Big) \\
&\leq C_1^{L-\ell}\frac{2}{\gamma}\Big(B_{\ell,d}^{\alpha,\beta}B_x C_1^{L} + 1 + \alpha\beta LC_1^{L}(\sqrt{d} + B_{\ell,d}^{\alpha,\beta}B_x)\Big)\|\Delta\boldsymbol{\theta}\|_2 \\
&\leq C_1^{L-\ell}\frac{2}{\gamma}\Big((\alpha\beta L + 1)(\sqrt{d} + B_{\ell,d}^{\alpha,\beta}B_x)C_1^{L} + 1\Big)\|\Delta\boldsymbol{\theta}\|_2 \\
&= C_1^{L-\ell}\frac{2}{\gamma}\Big((\alpha\beta L + 1)C_1^{L}C_2 + 1\Big)\|\Delta\boldsymbol{\theta}\|_2.
\end{aligned} \tag{158}$$

I.5 Proof of Lemma 26

By the definition of $\mathbf{G}^{(\ell)}$, we have

$$\begin{aligned}\|\mathbf{G}^{(\ell)}\|_2 &= \frac{1}{m}\left\|\beta\cdot[\alpha\mathbf{L}\mathbf{H}^{(\ell-1)}+(1-\alpha)\mathbf{X}]^\top\mathbf{D}^{(\ell)}\odot\sigma'(\boldsymbol{Z}^{(\ell)})\right\|_2\\ &\le \frac{1}{m}\beta\left\|[\alpha\mathbf{L}\mathbf{H}^{(\ell-1)}+(1-\alpha)\mathbf{X}]^\top\mathbf{D}^{(\ell)}\right\|_2\\ &\le \beta\Big(\alpha\sqrt{d}h_{\max}^{(\ell-1)}+(1-\alpha)B_x\Big)d_{\max}^{(\ell)}.\end{aligned} \tag{159}$$

Plugging in the results from Lemma 22 and Lemma 25, we have

$$\begin{aligned}\|\mathbf{G}^{(\ell)}\|_2 &\le \frac{2}{\gamma}\beta B_x\Big(\alpha\sqrt{d}\big((1-\alpha)(\ell-1)B_w^\beta+1\big)\cdot C_1^{\ell-1}+(1-\alpha)\Big)C_1^{L-\ell+1}\\ &\le \frac{2}{\gamma}\beta B_x\big((1-\alpha)\ell+\alpha\sqrt{d}\big)C_1^L \le \frac{2}{\gamma}\beta C_1^L C_2,\end{aligned} \tag{160}$$

where $C_2=\sqrt{d}+B_{\ell,d}^{\alpha,\beta}B_x$. Similarly, by the definition of $\mathbf{G}^{(L+1)}$, we have

$$\|\mathbf{G}^{(L+1)}\|_2 \le \frac{2}{\gamma}h_{\max}^{(L)} \le \frac{2}{\gamma}C_1^L C_2. \tag{161}$$

Therefore, we have

$$\sum_{\ell=1}^{L+1}\|\mathbf{G}^{(\ell)}\|_2 \le \frac{2}{\gamma}(L+1)\beta C_1^L C_2. \tag{162}$$

Furthermore, we can upper bound the difference between gradient for $\ell\in[L]$ as

$$\begin{aligned}\|\mathbf{G}^{(\ell)}-\tilde{\mathbf{G}}^{(\ell)}\|_2 &= \frac{1}{m}\left\|\beta[\alpha\mathbf{L}\mathbf{H}^{(\ell-1)}+(1-\alpha)\mathbf{X}]^\top\mathbf{D}^{(\ell+1)}\odot\sigma'(\mathbf{Z}^{(\ell)})-\beta[\alpha\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}+(1-\alpha)\mathbf{X}]^\top\tilde{\mathbf{D}}^{(\ell+1)}\odot\sigma'(\tilde{\mathbf{Z}}^{(\ell)})\right\|_2\\ &\underset{(a)}{\le} \frac{1}{m}\beta\|[\alpha\mathbf{L}\mathbf{H}^{(\ell-1)}+(1-\alpha)\mathbf{X}]^\top\mathbf{D}^{(\ell+1)}-[\alpha\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}+(1-\alpha)\mathbf{X}]^\top\mathbf{D}^{(\ell+1)}\|_2\\ &\quad+\frac{1}{m}\beta\|[\alpha\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}+(1-\alpha)\mathbf{X}]^\top\mathbf{D}^{(\ell+1)}-[\alpha\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}+(1-\alpha)\mathbf{X}]^\top\tilde{\mathbf{D}}^{(\ell+1)}\|_2\\ &\quad+\frac{1}{m}\beta\|[\alpha\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}+(1-\alpha)\mathbf{X}]^\top\tilde{\mathbf{D}}^{(\ell+1)}\odot\big(\sigma'(\mathbf{Z}^{(\ell)})-\sigma'(\tilde{\mathbf{Z}}^{(\ell)})\big)\|_2\\ &\underset{(b)}{\le} \beta\Big(\alpha\max_i\|[[\mathbf{L}(\mathbf{H}^{(\ell-1)}-\tilde{\mathbf{H}}^{(\ell-1)})]^\top\mathbf{D}^{(\ell+1)}]_{i,:}\|_2\\ &\quad+\max_i\|[[\alpha\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}+(1-\alpha)\mathbf{X}]^\top(\mathbf{D}^{(\ell+1)}-\tilde{\mathbf{D}}^{(\ell+1)})]_{i,:}\|_2+\\ &\quad+\max_i\|[[\alpha\mathbf{L}\tilde{\mathbf{H}}^{(\ell-1)}+(1-\alpha)\mathbf{X}]^\top\tilde{\mathbf{D}}^{(\ell+1)}]_{i,:}\|_2\max_i\|[\mathbf{Z}^{(\ell)}-\tilde{\mathbf{Z}}^{(\ell)}]_{i,:}\|_2\Big)\\ &\le \beta\Big(\underbrace{\alpha\sqrt{d}\Delta h_{\max}^{(\ell-1)}d_{\max}^{(\ell+1)}}_{(A)}+\underbrace{(\alpha\sqrt{d}h_{\max}^{(\ell-1)}+(1-\alpha)B_x)\Delta d_{\max}^{(\ell+1)}}_{(B)}\\ &\quad+\underbrace{\big(\alpha\sqrt{d}h_{\max}^{(\ell-1)}+(1-\alpha)B_x\big)d_{\max}^{(\ell+1)}\Delta z_{\max}^{(\ell)}}_{(C)}\Big),\end{aligned} \tag{163}$$

where inequality (a) and (b) is due to the gradient of ReLU activation is element-wise either 0 or 1. We can bound (A) as

$$\alpha\sqrt{d}\Delta h_{\max}^{(\ell-1)}d_{\max}^{(\ell+1)} \le \alpha\beta\sqrt{d}\frac{2}{\gamma}C_1^L C_2\|\Delta\boldsymbol{\theta}\|_2. \tag{164}$$

To upper bound (B) and (C), let first consider the upper bound of the following term

$$\begin{aligned}\alpha\sqrt{d}h_{\max}^{(\ell-1)}+(1-\alpha)B_x &\le \alpha\sqrt{d}\Big((1-\alpha)(\ell-1)B_w^\beta+1\Big)B_xC_1^{\ell-1}+(1-\alpha)B_x\\ &\le \alpha\sqrt{d}\Big((1-\alpha)\ell B_w^\beta+1\Big)B_xC_1^{\ell-1}\\ &\le \alpha\sqrt{d}C_1^{\ell-1}C_2.\end{aligned} \tag{165}$$

Therefore, we can bound (B) as

$$\Big(\alpha\sqrt{d}h_{\max}^{(\ell-1)} + (1-\alpha)B_x\Big)\Delta d_{\max}^{(\ell+1)} \leq \alpha\sqrt{d}C_1^{\ell-1}C_2 \cdot C_1^{L-\ell}\frac{2}{\gamma}\Big((\alpha\beta L+1)C_1^L C_2 + 1\Big)\|\Delta\boldsymbol{\theta}\|_2$$
$$\leq \alpha\sqrt{d}\frac{2}{\gamma}C_1^L C_2\Big((\alpha\beta L+1)C_1^L C_2 + 1\Big)\|\Delta\boldsymbol{\theta}\|_2, \tag{166}$$

and we can bound (C) by

$$\big(\alpha\sqrt{d}h_{\max}^{(\ell-1)} + (1-\alpha)B_x\big)d_{\max}^{(\ell+1)}\Delta z_{\max}^{(\ell)} \leq \alpha\sqrt{d}C_1^{\ell-1}C_2 \cdot \frac{2}{\gamma}C_1^{L-\ell}\cdot\beta C_1^{\ell-1}C_2\|\Delta\boldsymbol{\theta}\|_2$$
$$\leq \alpha\beta\sqrt{d}\frac{2}{\gamma}(C_1^L C_2)^2\|\Delta\boldsymbol{\theta}\|_2. \tag{167}$$

By combining result above, we have

$$\|\mathbf{G}^{(\ell)} - \tilde{\mathbf{G}}^{(\ell)}\|_2 \leq \alpha\beta\frac{2}{\gamma}\sqrt{d}C_1^L C_2\Big(2\beta + (\alpha\beta L+1)C_1^L C_2\Big). \tag{168}$$

Similarly, we can upper bound the difference between gradient for the weight of the binary classifier as

$$\|\mathbf{G}^{(L+1)} - \tilde{\mathbf{G}}^{(L+1)}\|_2 \leq \frac{2}{\gamma}\Delta h_{\max}^{(L)} \leq \beta\frac{2}{\gamma}C_1^L C_2\|\Delta\boldsymbol{\theta}\|_2. \tag{169}$$

which is upper bound by the right hand side of the previous equation. Therefore, we have

$$\sum_{\ell=1}^{L+1}\|\mathbf{G}^{(\ell)} - \tilde{\mathbf{G}}^{(\ell)}\|_2 \leq \alpha\beta\frac{2}{\gamma}(L+1)\sqrt{d}C_1^L C_2\Big(2\beta + (\alpha\beta L+1)C_1^L C_2\Big). \tag{170}$$

J Generalization bound for DGCN

In the following, we provide proof of the generalization bound of DGCN in Theorem 5. Recall that the update rule of DGCN is defined as

$$\mathbf{Z} = \sum_{\ell=1}^{L}\alpha_\ell\mathbf{H}^{(\ell)},\ \mathbf{H}^{(\ell)} = \mathbf{L}^\ell\mathbf{X}(\beta_\ell\mathbf{W}^{(\ell)} + (1-\beta_\ell)\mathbf{I}). \tag{171}$$

The training of DGCN is an empirical risk minimization with respect to a set of parameters $\boldsymbol{\theta} = \{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L)}, \mathbf{v}\}$, i.e.,

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m}\Phi_\gamma(-p(f(\mathbf{z}_i), y_i)),\ f(\mathbf{z}_i) = \tilde{\sigma}(\mathbf{v}^\top\mathbf{z}_i), \tag{172}$$

where $\mathbf{h}_i^{(L)}$ is the node representation of the ith node at the final layer, $f(\mathbf{h}_i^{(L)})$ is the predicted label for the ith node, $\tilde{\sigma}(x) = \frac{1}{\exp(-x)+1}$ is the sigmoid function, and loss function $\Phi_\gamma(-p(z,y))$ is $\frac{2}{\gamma}$-Lipschitz continuous with respect to its first input z. For simplification, let $\text{Loss}(z,y)$ denote $\Phi_\gamma(-p(z,y))$ in the proof.

For analysis purpose, we suppose α_ℓ and β_ℓ are hyper-parameters that are pre-selected before training and fixed during the training phase. However, in practice, these two parameters are tuned during the training phase.

To establish the generalization of DGCN as stated in Theorem 3, we utilize the result on transductive uniform stability from [16] (Theorem 8 in Appendix F). Then, in Lemma 27, we derive the uniform stability constant for DGCN, i.e., ϵ_{DGCN}.

Lemma 27. *The uniform stability constant for DGCN is computed as* $\epsilon_{\texttt{DGCN}} = \frac{2\eta\rho_f G_f}{m}\sum_{t=1}^{T}(1+\eta L_F)^{t-1}$ *where*

$$\rho_f = (\sqrt{d})^L B_x,\ G_f = \frac{2}{\gamma}(L+1)(\sqrt{d})^L B_x,$$
$$L_f = \frac{2}{\gamma}(L+1)(\sqrt{d})^L B_x\Big((\sqrt{d})^L B_x\max\{1, B_w\} + 1\Big). \tag{173}$$

By plugging the result in Lemma 27 back to Theorem 8, we establish the generalization bound for DGCN.

The key idea of the proof is to decompose the change of the network output into two terms (in Lemma 5) which depend on

- (Lemma 28) The maximum change of node representation $\Delta z_{\max}^{(\ell)} = \max_i \|[\mathbf{Z} - \tilde{\mathbf{Z}}]_{i,:}\|_2$,
- (Lemma 29) The maximum node representation $z_{\max}^{(\ell)} = \max_i \|[\mathbf{Z}]_{i,:}\|_2$.

Lemma 28 (Upper bound of $z_{\max}$ for **DGCN**)**.** *Let suppose Assumption 1 hold. Then, the maximum node embeddings for any node at the ℓth layer is bounded by*

$$z_{\max} \leq (\sqrt{d})^L B_x \max\{1, B_w\}. \tag{174}$$

Lemma 29 (Upper bound of $\Delta z_{\max}$ for **DGCN**)**.** *Let suppose Assumption 1 hold. Then, the maximum change between the node embeddings on two different set of weight parameters for any node at the ℓth layer is bounded by*

$$\Delta z_{\max} \leq (\sqrt{d})^L B_x (\|\Delta \mathbf{W}^{(1)}\|_2 + \ldots + \|\Delta \mathbf{W}^{(L)}\|_2), \tag{175}$$

where $\Delta \mathbf{W}^{(\ell)} = \mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)}$.

Then, in Lemma 30, we decompose the change of the model parameters into two terms which depend on

- The change of gradient with respect to the ℓth layer weight matrix $\|\Delta \mathbf{G}^{(\ell)}\|_2$,
- The gradient with respect to the ℓth layer weight matrix $\|\mathbf{G}^{(\ell)}\|_2$,

where $\|\Delta \mathbf{G}^{(\ell)}\|_2$ reflect the smoothness of APPNP model and $\|\mathbf{G}^{(\ell)}\|_2$ correspond the upper bound of gradient.

Lemma 30 (Upper bound of $\|\mathbf{G}^{(\ell)}\|_2, \|\Delta \mathbf{G}^{(\ell)}\|_2$ for **DGCN**)**.** *Let suppose Assumption 1 hold. Then, the gradient and the maximum change between gradients on two different set of weight parameters are bounded by*

$$\begin{aligned} &\sum_{\ell=1}^{L+1} \|\mathbf{G}^{(\ell)}\|_2 \leq \frac{2}{\gamma}(L+1)(\sqrt{d})^L B_x, \\ &\sum_{\ell=1}^{L+1} \|\Delta \mathbf{G}^{(\ell)}\|_2 \leq \frac{2}{\gamma}(L+1)(\sqrt{d})^L B_x \Big((\sqrt{d})^L B_x \max\{1, B_w\} + 1) \Big) \|\Delta \boldsymbol{\theta}\|_2, \end{aligned} \tag{176}$$

where $\|\Delta \boldsymbol{\theta}\|_2 = \|\mathbf{v} - \tilde{\mathbf{v}}\|_2 + \sum_{\ell-1}^{L} \|\mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)}\|_2$ *denotes the change of two set of parameters.*

Equipped with above intermediate results, we now proceed to prove Lemma 27.

Proof. Recall that our goal is to explore the impact of different GCN structures on the uniform stability constant ϵ_{DGCN}, which is a function of ρ_f, G_f, and L_f. By Lemma 29, we know that the function f is ρ_f-Lipschitz continuous, with $\rho_f = (\sqrt{d})^L B_x$. By Lemma 30, we know the function f is L_f-smoothness, and the gradient of each weight matrices is bounded by G_f, with

$$G_f \leq \frac{2}{\gamma}(L+1)(\sqrt{d})^L B_x, \; L_f \leq \frac{2}{\gamma}(L+1)(\sqrt{d})^L B_x \Big((\sqrt{d})^L B_x \max\{1, B_w\} + 1) \Big). \tag{177}$$

By plugging ϵ_{DGCN} into Theorem 3, we obtain the generalization bound of DGCN.

□

J.1 Proof of Lemma 28

By the definition of $z_{\max}$, we have

$$
\begin{aligned}
z_{\max} &= \max_i \left\| \left[\sum_{\ell=1}^{L} \alpha_\ell \mathbf{L}^\ell \mathbf{X}\big(\beta_\ell \mathbf{W}^{(\ell)} + (1-\beta_\ell)\mathbf{I}\big) \right]_{i,:} \right\|_2 \\
&\leq \sum_{\ell=1}^{L} \alpha_\ell \max_i \|[\mathbf{L}^\ell \mathbf{X}\big(\beta_\ell \mathbf{W}^{(\ell)} + (1-\beta_\ell)\mathbf{I}\big)]_{i,:}\|_2 \\
&\underset{(a)}{\leq} \sum_{\ell=1}^{L} \alpha_\ell (\sqrt{d})^\ell B_x \big(\beta_\ell B_w + (1-\beta_\ell)\big) \\
&\underset{(b)}{\leq} (\sqrt{d})^L B_x \max\{1, B_w\},
\end{aligned}
\tag{178}
$$

where inequality (a) follows from Lemma 31 and inequality (b) is due to the fact that $\sum_{\ell=1}^{L} \alpha_\ell = 1$ and $\alpha_\ell \in [0,1]$.

J.2 Proof of Lemma 29

By the definition of $\Delta z_{\max}$, we have

$$
\begin{aligned}
\Delta z_{\max} &= \max_i \|[\mathbf{Z} - \tilde{\mathbf{Z}}]_{i,:}\|_2 \\
&= \max_i \left\| \Big[\sum_{\ell=1}^{L} \alpha_\ell \mathbf{L}^\ell \mathbf{X}\big(\beta_\ell \mathbf{W}^{(\ell)} + (1-\beta_\ell)\mathbf{I}\big) - \sum_{\ell=1}^{L} \alpha_\ell \mathbf{L}^\ell \mathbf{X}\big(\beta_\ell \tilde{\mathbf{W}}^{(\ell)} + (1-\beta_\ell)\mathbf{I}\big) \Big]_{i,:} \right\|_2 \\
&\leq \sum_{\ell=1}^{L} \alpha_\ell \max_i \left\| \Big[\mathbf{L}^\ell \mathbf{X}\big(\beta_\ell \mathbf{W}^{(\ell)} + (1-\beta_\ell)\mathbf{I}\big) - \mathbf{L}^\ell \mathbf{X}\big(\beta_\ell \tilde{\mathbf{W}}^{(\ell)} + (1-\beta_\ell)\mathbf{I}\big) \Big]_{i,:} \right\|_2 \\
&= \sum_{\ell=1}^{L} \alpha_\ell \beta_\ell \max_i \left\| \Big[\mathbf{L}^\ell \mathbf{X}(\mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)}) \Big]_{i,:} \right\|_2 \\
&\underset{(a)}{\leq} \sum_{\ell=1}^{L} \alpha_\ell \beta_\ell (\sqrt{d})^\ell B_x \|\Delta \mathbf{W}^{(\ell)}\|_2 \\
&\leq (\sqrt{d})^L B_x \Big(\|\Delta \mathbf{W}^{(1)}\|_2 + \ldots + \|\Delta \mathbf{W}^{(L)}\|_2 \Big),
\end{aligned}
\tag{179}
$$

where inequality (a) follows from Lemma 31 and inequality (b) is due to the fact that $\sum_{\ell=1}^{L} \alpha_\ell = 1$ and $\alpha_\ell \in [0,1]$.

J.3 Proof of Lemma 30

Recall that $\mathbf{G}^{(\ell)}$ is defined as the gradient of loss with respect to $\mathbf{W}^{(\ell)}$, which can be formulated as

$$\begin{aligned}
\|\mathbf{G}^{(\ell)}\|_2 &= \Big\|\frac{1}{m}\sum_{i=1}^{m}\frac{\partial \text{Loss}(f(\mathbf{z}_i), y_i)}{\partial \mathbf{z}_i}\frac{\partial \mathbf{z}_i}{\partial \mathbf{h}_i^{(\ell)}}\frac{\partial \mathbf{h}_i^{(\ell)}}{\partial \mathbf{W}^{(\ell)}}\Big\|_2 \\
&\leq \max_i \Big\|\frac{\partial \text{Loss}(f(\mathbf{z}_i), y_i)}{\partial \mathbf{z}_i}\frac{\partial \mathbf{z}_i}{\partial \mathbf{h}_i^{(\ell)}}\frac{\partial \mathbf{h}_i^{(\ell)}}{\partial \mathbf{W}^{(\ell)}}\Big\|_2 \\
&\leq \alpha_\ell \beta_\ell (\sqrt{d})^\ell B_x \max_i \Big\|\frac{\partial \text{Loss}(f(\mathbf{z}_i), y_i)}{\partial \mathbf{z}_i}\Big\|_2 \\
&\underset{(a)}{\leq} \frac{2}{\gamma}\alpha_\ell \beta_\ell (\sqrt{d})^\ell B_x \\
&\leq \frac{2}{\gamma}(\sqrt{d})^L B_x,
\end{aligned} \tag{180}$$

where (a) is due to the fact that loss function is $\frac{2}{\gamma}$-Lipschitz conitnous.

Besides, recall that $\mathbf{G}^{(L+1)}$ denotes the gradient with respect to the weight of binary classifier. Therefore, we have

$$\|\mathbf{G}^{(L+1)}\|_2 \leq \frac{2}{\gamma}h_{\max}^{(L)} \underset{(a)}{\leq} \frac{2}{\gamma}(\sqrt{d})^L B_x, \tag{181}$$

where inequality (a) follows from Lemma 29.

Therefore, combining the results above, we have

$$\sum_{\ell=1}^{L+1}\|\mathbf{G}^{(\ell)}\|_2 \leq \frac{2}{\gamma}(L+1)(\sqrt{d})^L B_x. \tag{182}$$

Furthermore, by the definition of $\|\mathbf{G}^{(\ell)} - \tilde{\mathbf{G}}^{(\ell)}\|_2$, we have

$$\begin{aligned}
\|\mathbf{G}^{(\ell)} - \tilde{\mathbf{G}}^{(\ell)}\|_2 &\leq \frac{2}{\gamma}\max_i \|\frac{\partial f(\mathbf{z}_i)}{\partial \mathbf{z}_i}\frac{\partial \mathbf{z}_i}{\partial \mathbf{W}} - \frac{\partial f(\tilde{\mathbf{z}}_i)}{\partial \tilde{\mathbf{z}}_i}\frac{\partial \tilde{\mathbf{z}}_i}{\partial \tilde{\mathbf{W}}}\|_2 \\
&= \frac{2}{\gamma}\alpha_\ell \beta_\ell (\sqrt{d})^\ell B_x \Big\|\frac{\partial f(\mathbf{z}_i)}{\partial \mathbf{z}_i} - \frac{\partial f(\tilde{\mathbf{z}}_i)}{\partial \tilde{\mathbf{z}}_i}\Big\|_2 \\
&\leq \frac{2}{\gamma}\alpha_\ell \beta_\ell B_x (\sqrt{d})^\ell \Big(\Delta z_{\max} + (z_{\max}+1)\|\Delta \mathbf{v}\|_2\Big) \\
&\leq \frac{2}{\gamma}(\sqrt{d})^L B_x \Big((\sqrt{d})^L B_x \max\{1, B_w\} + 1)\Big)\|\Delta \boldsymbol{\theta}\|_2.
\end{aligned} \tag{183}$$

Similarly, we can upper bound the difference between gradient for the weight of the binary classifier as

$$\begin{aligned}
\|\mathbf{G}^{(L+1)} - \tilde{\mathbf{G}}^{(L+1)}\|_2 &\leq \frac{2}{\gamma}\Delta h_{\max}^{(L)} \\
&\leq \frac{2}{\gamma}(\sqrt{d})^L B_x B_w,
\end{aligned} \tag{184}$$

which is bounded by the right hand side of the last equation. Therefore, we have

$$\sum_{\ell=1}^{L+1}\|\mathbf{G}^{(\ell)} - \tilde{\mathbf{G}}^{(\ell)}\|_2 \leq \frac{2}{\gamma}(L+1)(\sqrt{d})^L B_x \Big((\sqrt{d})^L B_x \max\{1, B_w\} + 1)\Big)\|\Delta \boldsymbol{\theta}\|_2. \tag{185}$$

K Omitted proofs of useful lemmas

K.1 Proof of Lemma 1

Let $f_k(\boldsymbol{\theta})$ denote applying function f with parameter $\boldsymbol{\theta}$ on the kth data point and $\boldsymbol{\theta}_t$ as the weight parameter at the tth iteration. Because function $f_k(\boldsymbol{\theta})$ is ρ_f-Lipschitz with respect to parameter $\boldsymbol{\theta}$, we

can bound the difference between model output by the difference between parameters $\boldsymbol{\theta}_T^{ij}, \boldsymbol{\theta}_T$, i.e.,

$$\epsilon = \max_k |f_k(\boldsymbol{\theta}_T^{ij}) - f_k(\boldsymbol{\theta}_T)| \leq \rho_f \|\boldsymbol{\theta}_T^{ij} - \boldsymbol{\theta}_T\|_2. \tag{186}$$

Let $\nabla f(\boldsymbol{\theta}) = \frac{1}{m}\sum_{k=1}^m \nabla f_k(\boldsymbol{\theta})$ and $\nabla f(\boldsymbol{\theta}^{ij}) = \frac{1}{m}\sum_{k=1}^m \nabla f_k(\boldsymbol{\theta}^{ij})$ denote the full-batch gradient computed on the original and perturbed dataset respectively.

Recall that both models have the same initialization $\boldsymbol{\theta}_0 = \boldsymbol{\theta}_0^{ij}$. At each iteration the full-batch gradient are computed on $(m-1)$ data points that identical in two dataset, and only 1 data point that are different in two dataset. Therefore, we can bound the change of model parameters $\boldsymbol{\theta}_t^{ij}, \boldsymbol{\theta}_t$ after one gradient update step as

$$\begin{aligned}\|\boldsymbol{\theta}_{t+1}^{ij} - \boldsymbol{\theta}_{t+1}\|_2 &\leq \Big(1 + (1 - \frac{1}{m})\eta L_f\Big)\|\boldsymbol{\theta}_t^{ij} - \boldsymbol{\theta}_t\|_2 + \frac{2\eta G_f}{m} \\ &\leq (1 + \eta L_f)\|\boldsymbol{\theta}_t^{ij} - \boldsymbol{\theta}_t\|_2 + \frac{2\eta G_f}{m}.\end{aligned} \tag{187}$$

Then after T iterations, we can bound the different between two parameters as

$$\|\boldsymbol{\theta}_T^{ij} - \boldsymbol{\theta}_T\|_2 \leq \frac{2\eta G_f}{m}\sum_{t=1}^T (1 + \eta L_F)^{t-1}. \tag{188}$$

By plugging the result back to Eq. 186, we have

$$\epsilon = \frac{2\eta\rho_f G_f}{m}\sum_{t=1}^T (1 + \eta L_F)^{t-1}. \tag{189}$$

K.2 Upper bound on the ℓ_1-norm of Laplacian matrix

Lemma 31 (Lemma A.3 in [35])**.** *Let* $\mathbf{A}$ *denote the adjacency matrix of a self-connected undirected graph* $\mathcal{G}(\mathcal{V}, \mathcal{E})$*, and* $\mathbf{D}$ *denote its corresponding degree matrix, and* d *denote the maximum number of node degree. We define the graph Laplacian matrix as* $\mathbf{L} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$*. Then we have*

$$\max_i \|[\mathbf{L}]_{i,:}\|_2 \leq \sqrt{d}. \tag{190}$$

Proof. By the definition of Laplacian matrix, we know that the ith row and jth column of $\mathbf{L}$ is defined as $L_{i,j} = \frac{1}{\sqrt{\deg(i)}\sqrt{\deg(j)}}$. Therefore, we have

$$\begin{aligned}\max_i \sum_{j=1}^N L_{i,j} &= \max_i \sum_{j\in\mathcal{N}(i)} \frac{1}{\sqrt{\deg(i)}\sqrt{\deg(j)}} \\ &\underset{(a)}{\leq} \max_i \sum_{j\in\mathcal{N}(i)} \frac{1}{\sqrt{\deg(i)}} \\ &= \max_i \sqrt{\deg(i)} \leq \sqrt{d},\end{aligned} \tag{191}$$

where (a) is due to $\frac{1}{\sqrt{\deg(i)}} \leq 1$ for any node i.

□