Statistical Foundations for Learning on Graphs

by

Aseem Baranwal

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2024

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Xavier Bresson Associate Professor, Dept. of Computer Science National University of Singapore
Supervisors:	Kimon Fountoulakis Associate Professor, Dept. of Computer Science
	Aukosh Jagannath Assistant Professor, Dept. of Statistics and Actuarial Science, Dept. of Applied Mathematics, Dept. of Computer Science
Internal Member:	Gautam Kamath Assistant Professor, Dept. of Computer Science
Internal Member:	Yaoliang Yu Associate Professor, Dept. of Computer Science
Internal-External Member:	Stephen A. Vavasis Professor, Dept. of Combinatorics and Optimization

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Research presented in Chapters 3, 4 and 5: Dr. Kimon Fountoulakis was the primary investigator of the Natural Sciences and Engineering Research Council of Canada (NSERC) grant [RGPIN-2019-04067, DGECR-2019-00147] and Dr. Aukosh Jagannath was the primary investigator of the Natural Sciences and Engineering Research Council of Canada (NSERC) grant [RGPIN-2020-04597, DGECR-2020-00199]. These two grants supported this research work. Both Dr. Fountoulakis and Dr. Jagannath are co-authors on publications related to these Chapters 3 and 5 [BFJ21b, BFJ23a, BFJ23b]. As the lead author of these publications, I was responsible for the design of the relevant statistical framework and the analysis of various methods in that framework, along with suitable experiments that supplement the results. Dr. Kimon Fountoulakis, Dr. Aukosh Jagannath, Shenghao Yang and Dr. Amit Levi are co-authors on the publication related to Chapter 4 [FLY$^+$23], where my primary contribution was to prove an impossibility result regarding graph attention mechanisms.

Research Presented in Chapter 6: Robert Wang and Dr. Kimon Fountoulakis are co-authors on the publication related to this chapter [WBF24], where I present the work as subsequent relevant work instead of a part of the main contributions of this thesis.

Abstract

Graph Neural Network(s) (GNN) are one of the most popular architectures used to solve classification problems on data where entities have attribute information accompanied by relational information. Among them, Graph Convolutional Network(s) (GCN) and Graph Attention Network(s) (GAT) are two of the most popular GNN architectures. In this thesis, I present a statistical framework for understanding node classification on feature-rich relational data. First, I use the framework to study the generalization error and the effects of existing neural network architectures, namely, graph convolutions and graph attention on the Contextual Stochastic Block Model (CSBM) in the regime where the average degree of a node is $\Omega(\log^2 n)$ in the number of nodes n. Second, I propose a notion of asymptotic local optimality for node classification tasks and design a GNN architecture that is provably optimal in this notion, for the sparse regime, i.e., average degree $O(1)$.

In the first part, I present a rigorous theoretical understanding of the effects of graph convolutions in neural networks through the node classification problem of a non-linearly separable Gaussian mixture model coupled with a stochastic block model. First, I identify two quantities corresponding to the signal from the two sources of information: the graph, and the node features, followed by a result that shows that a single graph convolution expands the regime of the distance between the means where multi-layer networks can classify the data by a factor of up to $1/\sqrt{\mathbb{E}\deg}$, where $\mathbb{E}\deg$ denotes the expected degree of a node. Second, I show that with a slightly stronger graph density, two graph convolutions improve this factor to up to $1/\sqrt{n}$, where n is the number of nodes in the graph. This set of results provides both theoretical and empirical insights into the performance of graph convolutions placed in different combinations among the layers of a neural network, concluding that the performance is mutually similar for all combinations of the placement.

In the second part, the analysis of graph attention is provided, where the main result states that in a well-defined “hard” regime, every attention mechanism fails to distinguish the intra-class edges from the inter-class edges. In addition, if the signal in the node attributes is sufficiently weak, graph attention convolution cannot perfectly classify the nodes even if the intra-class edges are separable from the inter-class edges.

In the third part, I study the node classification problem on feature-decorated graphs in the sparse setting, i.e., when the expected degree of a node is $O(1)$ in the number of nodes, in the fixed-dimensional asymptotic regime, i.e., the dimension of the feature data is fixed while the number of nodes is large. Such graphs are typically known to be locally tree-like. Here, I introduce a notion of Bayes optimality for node classification tasks, called asymptotic local Bayes optimality, and compute the optimal classifier according to this criterion for a fairly general statistical data model with arbitrary distributions of the node features and edge connectivity. The optimal classifier is implementable using a message-passing graph neural network architecture. This is followed by a result that precisely computes the generalization error of this optimal classifier, and compares its performance statistically against existing learning methods on a well-studied data model with naturally identifiable signal-to-noise ratios (SNRs). We find that the optimal message-passing architecture interpolates between a standard MLP in the regime of low graph signal and a typical graph convolutional layer in the regime of high graph signal. Furthermore, I provide a corresponding non-asymptotic result that demonstrates the practical potential of the asymptotically optimal classifier.

All the results are supplemented with extensive experiments on both synthetic and real-world data to illustrate the main theorems. The code is open-sourced on GitHub.

Acknowledgements

I always thought I would not take names in my thesis acknowledgements, but it's hard not to mention some crucial people I've had with me during this journey.

Firstly, thanks to my partner in every sense, Pranika, for keeping us close during our long-distance relationship, for handling the consequences of my stupidity several times, and for making sure I kept my motivation as high as when I started and my mental and physical health the best possible. Thanks to her for completing the meaning of life for me. She is everything that was missing from my life.

Secondly, thanks to my younger brother, Akhil, who always lost to me at `Future Cop: LAPD`, and yet, I look up to him for things that actually matter. You're the best.

Thirdly, I know that I wouldn't have been able to do this without the unending love and support of my friends. At Waterloo, I was lucky to find some of the most beautiful friendships. My roommates and closest friends: Shubhankar, Aaishwarya, Himanshi, Rishav, Manav, Parth, Gaurav and Khadija who saw the best and the worst of me through the whole journey, played all kinds of board games and video games with me, went on several camping trips and vacations with me, and stood with me through all the happy and sad moments. My friends Shenghao, Artur, Justin, Chendi, Marc, Clara, Amine and Tosca made conducting research and coming to the campus quite fun.

Finally, my advisors Kimon and Aukosh who always had my best interests at heart. Being their first PhD student, I believe we will always have a special student-advisor relationship. I would also like to thank my dear committee members for a great discussion during my defence and for all their meaningful suggestions.

Dedication

To my parents, Sudha and Praveen, who raised me to be what I am, sacrificed their own dreams to cater to mine, taught me to think for myself, and fully supported my desire to travel halfway around the world to pursue my academic endeavours.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

a.s. almost surely 86, 88

CSBM Contextual Stochastic Block Model v, 9

GAT Graph Attention Network(s) v, 2–4, 6

GCN Graph Convolutional Network(s) v, xiv, 1, 2, 4, 6, 7, 51, 80, 86–88, 94, 107, 110

GNN Graph Neural Network(s) v, 2, 3, 113

iid independent and identically distributed 12, 17, 19, 22, 51, 53, 55, 58, 94, 104

Chapter 1

Introduction

A large amount of interesting data and the practical challenges associated with them are defined in the setting where entities have attributes as well as information about mutual relationships. Traditional classification models have been extended to capture such relational information through graphs [Ham20], where each node has individual attributes and the edges of the graph capture the relationships among the nodes. A variety of applications characterized by this type of graph-structured data include works in the areas of social analysis [BL11], recommendation systems [YHC+18], computer vision [MBM+17], study of the properties of chemical compounds [GSR+17, SGT+09], statistical physics [BKGB+20, BPL+16], and financial forensics [ZZY+17, WDC+19].

Various popular learning methods for relational data utilize graph convolutions [KW17], where the idea is to aggregate the attributes of the set of neighbours of a node instead of only using its attributes. Despite several empirical studies of various GCN-type models [CLB19, MLST22] that demonstrate that graph convolutions can improve the performance of traditional classification methods, there has been limited progress in the theoretical understanding of the benefits of graph convolutions in multi-layer networks in terms of improvement on node classification tasks.

1.1 Literature Review

1.1.1 Graph Convolutions

The capacity of a graph convolution for one-layer networks is studied in [BFJ21b], along with its out-of-distribution (OoD) generalization potential. A more recent work [WZYW22] formulates the node-level OoD problem and develops a learning method that facilitates GNNs to leverage invariance principles for prediction. In [GBG19], the authors utilize a propagation scheme based on personalized PageRank to construct a model that outperforms several GCN-like methods for semi-supervised classification. Through their algorithm (APPNP) they show that placing power iterations at the last layer of an MLP achieves state-of-the-art performance. As a byproduct of our results, we verify this empirical observation theoretically on a well-studied data model, the contextual stochastic block model (CSBM) [DSMM18].

To this end, we study the effects of graph convolutions in deeper layers of a multilayer network. For node classification tasks, we also study whether one can avoid using additional layers in the network design for the sole purpose of gathering information from neighbours that are farther away, by comparing the benefits of placing all convolutions in a single layer versus placing them in different layers.

1.1.2 Graph Attention

Graph convolution, usually defined using its spatial version, corresponds to averaging the features of a node with the features of its neighbours [KW17]. More broadly, graph convolution can refer to different variants arising from different (approximations of) graph spectral filters. Graph attention [VCC+18a] mechanisms augment this convolution by appropriately weighting the edges of a graph before spatially convolving the data. The weighting can be done using information from the given features for each node. Despite its wide adoption by practitioners [FL19, WZY+19, HFZ+20a] and its large academic impact, the number of works that rigorously study its effectiveness is quite limited.

Recently, the concept of attention for neural networks [BCB15, VSP+17] was transferred to GNNs [LZBT16, BL17, VCC+18a, LRK+19, PBHL20]. A few papers have attempted to understand the attention mechanism in [VCC+18a]. One work relevant to ours is [BAY22]. In this paper, the authors show that a node may fail to assign large edge weight to its most important neighbours due to a global ranking of nodes generated by the attention mechanism in [VCC+18a]. Another related work is [KTA19], which presents an empirical study of the ability of graph attention to generalize on larger, complex, and noisy graphs. In addition, in [HZC+19] the authors propose a different metric to generate the attention coefficients and show empirically that it has an advantage over the original GAT architecture.

Other related work to ours, which does not focus on graph attention, comes from the field of statistical learning on random data models. Random graphs and the stochastic block model have been traditionally used in clustering and community detection [Abb18, AFT$^+$18, Moo17]. Moreover, the works by [BVR17, DSMM18], which also rely on CSBM are focused on the fundamental limits of unsupervised learning. Of particular relevance is the work by [BFJ21a], which studies the performance of graph convolution on CSBM as a semi-supervised learning problem. Within the context of random graphs, [KBV21a] studies the approximation power of GNNs on random graphs. In [MLLK22a] the authors derive the generalization error of GNNs for graph classification and regression tasks. In [FLY$^+$23], the authors attempt to understand graph attention's capability for edge/node classification for different parameter regimes of CSBM.

Finally, there are a few related theoretical works on understanding the generalization and representation power of GNNs [CLB19, CPLM20, ZYZ$^+$20, XHLJ19, GJJ20, Lou20a, Lou20b]. For a recent survey in this direction see [Jeg22]. This section of the thesis is based on work done in [FLY$^+$23], where we take a statistical perspective, allowing us to characterize the precise performance of graph attention compared to graph convolution and no convolution for CSBM, to answer the particular questions imposed above.

1.1.3 Statistical Models

There exists a large amount of theoretical work on unsupervised learning for random graph models where node features are absent and only relational information is available [DKMZ11, Mas14, MNS18, MNS15a, AS15, ABH15, BLM15, DAM15, MS16, BMNN16, AS18, LCM19, KUK17, GRS22]. For a comprehensive survey, see [Abb18, Moo17].

Several theoretical and empirical works have also studied data models which have node features coupled with relational information. In [DSMM18, LS20], the authors explore the fundamental thresholds for weak recovery and community detection in the regime where the number of nodes scales with the dimension of data, and the average degree is constant. For the (primarily) empirical study of the semi-supervised node classification problem, see [SGT$^+$09, CZY11, GVB12, DV12, GFRS13, YML13, HYL17, JLL$^+$19, MDR19, CCH$^+$22, YHS$^+$21]. These papers provide good empirical insights into the merits of graph structure in the data. I complement these studies with theoretical results that explain the effects of graph convolutions in a multi-layer network.

Another relatively recent work [HZC$^+$20] proposes two graph smoothness metrics for measuring the benefits of graphical information, along with a new attention-based framework. In [FLY$^+$23], the authors provide a theoretical study of the graph attention mechanism (GAT) and identify the regimes where the attention mechanism is (or is not) beneficial to node classification tasks. Our study focuses on convolutions instead of attention-based mechanisms. Several other works study the expressive power and extrapolation of GNNs, along with the oversmoothing phenomenon (see, e.g., [BRH+21, XZL+21, OS20, LHW18]), however, our focus is to draw a comparison of the benefits and limitations of graph convolutions with those of a traditional MLP that does not utilize relational information.

These areas of research still lack theoretical guarantees that explain when and why graphical data, and in particular, graph convolutions, can boost traditional multi-layer networks to perform better on node classification tasks.

1.1.4 Message-Passing GNN Architectures

There has been a tremendous amount of work on GNN architecture design, where the most popular designs are based on a convolutional architecture, with each layer of the neural network performing a weighted convolution (averaging) operation with immediate neighbours, e.g., graph convolutional networks (GCN) [KW17, CWH+20] or graph attention networks (GAT) [VCC+18b]. These architectures are known to have several limitations regarding their expressive power (see, e.g., [LHW18, OS20, BRH+21, XZL+21, Ker22]).

An interesting line of research consists of both theoretical and empirical works that attempt to address these limitations by developing an understanding of GNN architectures within the scope of message-passing [RHXH20, LJZ+22, MLLK22b], as well as beyond it [MBHSL19, MSRR19, CVCB19]. For example, [XLT+18] propose an architecture with a technique called skip-connections, that flexibly leverages different ranges of neighbourhoods for each node to enable structure-awareness in node representations, [CWH+20] propose a modification of the vanilla GCN with an initial residual that effectively relieves the problem of oversmoothing [OS20], and [KBV21b] study the universality of structural GNNs in the large random graph limit. However, this area of research still lacks a clear understanding of optimality in the context of graph learning problems, making it hard to design architectures for which a well-defined notion of optimality can be theoretically justified.

Several works have studied traditional message-passing GNN architectures like GCN and GAT using the binary contextual stochastic block model, see for example, [BFJ21b, CCH+22, FLY+23, JMLV22, BFJ23a]. These analyses rely heavily on two assumptions: first, the graph is not too sparse, i.e., for a graph with n nodes, the expected degree of a node is $\Omega_n(\log^2 n)$, and second, the node features are modelled as a Gaussian mixture. The work by [WYJ+22] is of particular interest to us, where the authors take a Bayesian inference perspective to investigate the functions of non-linearity in GNNs for binary node classification. They characterize the max-a-posterior estimation of a node label given the features of itself and its immediate neighbours. A similar perspective to that of [WYJ+22] is discussed in [GSGG23], where the latter authors derive insights into the robustness-accuracy trade-off in GNNs for node classification.

In contrast to these inspiring works, I study the highly sparse regime for this part of the thesis, where the expected degree of a node is $O_n(1)$, and also consider nodes beyond the immediate neighbours at any fixed distance. (In fact, the non-asymptotic results allow distances of order $c \log n$ for small enough $c > 0$, see Section 5.5). Furthermore, my main result (Theorem 7) holds for a general multi-class statistical model with arbitrary continuous or discrete feature distributions and arbitrary edge-connectivity probabilities between all pairs of classes.

1.2 Overview of Contributions

The main results in this thesis are presented in three parts.

1.2.1 Effects of Graph Convolutions

The first part introduces a theoretical framework based on the contextual stochastic block model (see Section 2.2) that provides a statistical standard for benchmarking various GNN architectures for node classification tasks. Following the description of this framework, I present a rigorous study of the effects of graph convolutions on feature-rich relational data in the regime where the average node degree is $\omega(\log n)$ in the number of nodes n. I also recognize classification thresholds based on meaningful signals identified in the two sources of data: the node features and the node relationships. First, I develop an intuition for the main ideas using a simple binary CSBM (contextual stochastic block-model, see Section 2.2). The following results are for the binary model where a one-layer network is sufficient to classify the data.

1. I compare the relative performance of a graph convolution to the Bayes optimal classifier in the case where we do not have a graph. In the absence of relational information, I identify the threshold below which the data is not linearly separable with high probability, followed by computing the precise improvement in this threshold offered by a graph convolution. This result is formally presented in Theorem 1.
2. The second result is about the generalization of the output of the corresponding optimization procedure that runs on a training sample. In particular, I show that

the minimizer performs well on the classification of unseen out-of-distribution data. See Theorem 2 for a formal statement.

The insights from the study of this toy model motivate the study of a slightly more complicated variant, which I refer to as the XOR-CSBM (see Section 3.3). Here, one necessarily needs a network with at least two layers to classify the data. For this model, I study neural networks with up to three layers and show the improvement obtained by up to two graph convolutions. In particular, the following results are obtained:

1. A single graph convolution enables a multi-layer network to classify the nodes in a wider regime as compared to methods that do not utilize the graph, improving the threshold for the distance between the means of the features by a factor of up to $1/\sqrt{\mathbb{E}\deg}$. Furthermore, with a slightly denser graph, a multi-layer network with two graph convolutions can classify the data in an even wider regime, improving the threshold by a factor of up to $1/\sqrt{n}$, where n is the number of nodes in the graph. This is formalized in Theorem 4.
2. For multi-layer networks equipped with graph convolutions, the classification threshold is determined by the number of graph convolutions rather than the number of layers in the network. I study the improvement in the classification threshold obtained by placing graph convolutions in several combinations across the layers of a multi-layer network and find that the performance is mutually similar for all combinations with the same number of graph convolutions.

1.2.2 Analysis of Graph Attention

In this part, I present the main result for the "hard regime", i.e., when the distance between the means is small compared to the standard deviation, showing that *every* attention architecture fails to distinguish inter-class edges from intra-class edges with high probability (Theorem 5).

Moreover, for the original GAT architecture [VCC+18a], it can be shown that with high probability, most of the attention coefficients have uniform weights, similar to those of a simple GCN [KW17] (Theorem 6).

1.2.3 Optimality of Message-Passing

Next, I turn to the regime of very sparse graphs where the average node degree is $O(1)$, where I propose a notion of asymptotically local Bayes optimality. The same statistical data model is extended to arbitrary node features and edge-connectivity profiles among all pairs of classes to study this setting. The following results and findings are obtained:

1. Introduction to a family of GNN architectures that are asymptotically (in the number of nodes $n \to \infty$) Bayes optimal in a local sense for a general multi-class data model with arbitrary feature distributions. The optimality is stated precisely in Theorem 7.
2. Analysis of the architecture in the simpler two-class setting with Gaussian features, explicitly characterizing the generalization error in terms of the natural signal-to-noise ratio (SNR) in the data.
3. A comparative study against other learning methods analyzed using the same statistical model (see Theorems 8 and 9). There are two key insights from this study:
   - When the graph SNR is very low, the architecture reduces to a simple MLP that does not consider the graph at all, while if it is very high, the architecture reduces to a typical convolutional network that averages information from all nodes in the local neighbourhood. In the regime between the low and high SNRs, the architecture interpolates and performs better than both a simple MLP and a typical GCN.
   - If the signal in the graph is larger than a threshold, then a simple convolution can perform better than any method that does not utilize the graph.
4. In the non-asymptotic setting with a fixed number of nodes, we show that even for a logarithmic depth, the neighbourhoods of an overwhelming fraction of nodes are tree-like with high probability. Subsequently, we show that the optimal classifier in the non-asymptotic setting obtains an error close to that incurred by the optimal classifier in the asymptotic setting. This is formalized in Theorem 10.

All of the content in this thesis is based on the papers published during my Ph.D. with my supervisors and co-authors [BFJ21b, BFJ23a, FLY$^+$23, BFJ23b, WBF24]. All the results are demonstrated through extensive experiments on both synthetic and real data. The source code for all experiments is open-sourced and can be found on GitHub. The reader is referred to the GitHub account github.com/opallab/ for more details.

Chapter 2

Preliminaries and Background

2.1 Common Notation

In the rest of this thesis, vectors and matrices are usually denoted by lowercase and uppercase bold alphabets, respectively (such as $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{X} \in \mathbb{R}^{n\times d}$). $\mathbf{I}_n$ and $\mathbf{J}_n$ refer to the n-dimensional identity matrix and all-ones matrix, respectively, where the subscript is omitted if inferred from context. We write $[n] \overset{\text{def}}{=} \{1, 2, \ldots, n\}$. We use $\mathrm{Ber}(p)$ to denote the Bernoulli distribution, so $x \sim \mathrm{Ber}(p)$ means the random variable x takes value 1 with probability p and 0 with probability $1 - p$.

$G = (V, E)$ will denote a graph with the set of vertices $V(G)$ and edges $E(G)$. The adjacency matrix is denoted by $\mathbf{A} = \{a_{uv}\}_{u,v\in[n]}$, and the degree of a node u in G is denoted by $\mathbf{deg}(u, G) = \sum_{v\in[n]} a_{uv}$, where G is omitted if inferred from context.

Define the canonical graph distance metric (breadth-first distance / shortest distance) between two nodes u, v in an undirected, unweighted graph to be $d(u, v)$. For a node $u \in V(G)$, define $\eta_k(u) = \{v \in V(G) : d(u, v) \leq k\}$ to be the ball of radius k for the canonical graph distance metric. I will also use $N_k(u)$ to denote the set of vertices at the distance of exactly k from node u. Thus, $\eta_k(u) = \{u\} \cup_{j=1}^{k} N_k(u)$.

The all-ones vector is denoted by $\mathbf{1}$ and e_i denotes the i^{th} standard basis vector in $\mathbb{R}^n$. Given a vector $\mathbf{x} \in \mathbb{R}^n$, $\|\mathbf{x}\|$ denotes its Euclidean norm $\sqrt{\sum_{i=1}^{n} x_i^2}$. We use $\|\mathbf{x}\|_\infty$ to denote its infinity norm, $\max_{i=1}^{n} |x_i|$. For a matrix $\mathbf{M} \in \mathbb{R}^n$, we use $\|\mathbf{M}\|$ to denote its spectral norm, $\max_{\mathbf{x}\neq\mathbf{0},\|\mathbf{x}\|=1} \|\mathbf{M}\mathbf{x}\|$. We also make routine use of the spectral theorem, which says that if $\mathbf{M}$ is an $n \times n$ symmetric matrix, then it can be diagonalized with n orthogonal eigenvectors and real eigenvalues. In particular, there exist $\lambda_1, \lambda_2, \ldots \lambda_n \in$

$\mathbb{R}$ and orthonormal vectors $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n \in \mathbb{R}^n$ such that $\mathbf{M} = \sum_{i=1}^n \lambda_i \mathbf{w}_i \mathbf{w}_i^\top$. Note that when $\mathbf{M}$ is symmetric, $\|\mathbf{M}\| = \max_i |\lambda_i| = \max_{\mathbf{x}:\|\mathbf{x}\|=1} |x^\top \mathbf{M}\mathbf{x}|$. Finally, $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ denotes a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix Σ. For one-dimensional Gaussians, we use $\mathcal{N}(\mu, \sigma^2)$. For $X \sim \mathcal{N}(\mu, \sigma^2)$, we will frequently use the Gaussian tail bound: $\mathbf{Pr}\left[|X - \mu| > t\sigma\right] \leq \exp(-\frac{t^2}{2})$. We will usually denote by ϕ the sigmoid function $1/(1 + exp(x))$ while Φ will always stand for the CDF of the standard normal distribution.

2.2 Statistical Data Model

I work with the multi-class CSBM, where each node belongs to one of C different classes labelled $1, \ldots, C$, and the node features have arbitrary continuous or discrete distributions. This model with $C = 2$, along with a specialization to Gaussian features has been extensively studied in several works on (semi)-supervised node classification and unsupervised community detection, see, for example, [DSMM18, LS20, BFJ21b, WYJ$^+$22, FLY$^+$23, BFJ23a]. Informally, a CSBM consists of a coupling of a stochastic block model (SBM) [HLL83] with a mixture model where the components of the mixture have arbitrary distributions and are associated with the blocks of the SBM.

More formally, let n, d be positive integers such that n denotes the number of nodes and d denotes the dimension of the node features. Define $y_1, \ldots, y_n \in \{1, \ldots, C\}$ as the latent variables (class labels) to be inferred. Assume that the latent variables have a uniform prior, i.e., $y_u \sim \mathrm{Unif}(\{[C]\})$ for all u. For the relational part of the data, we have an undirected unweighted graph of n nodes, $G = (V, E)$ with adjacency matrix $\mathbf{A} = (a_{uv})_{u,v\in[n]} \sim \mathrm{SBM}(n, \mathbf{Q})$, where $\mathbf{Q} = \{q_{ij}\} \in [0,1]^{C\times C}$ is the edge-probability matrix, meaning that

$$\mathbf{Pr}(a_{uv} = 1 \mid y_u = i, y_v = j) = q_{ij}.$$

The node attributes, $\mathbf{X} \in \mathbb{R}^{n\times d}$ are sampled from a mixture of C arbitrary distributions, $\mathbb{P} = \{\mathbb{P}_i\}_{i\in[C]}$, where corresponding to the y_u, we have $\mathbf{X}_u \sim \mathbb{P}_{y_u}$ for all $u \in [n]$. For a feature-decorated graph $G = (\mathbf{A}, \mathbf{X}) = (\{a_{uv}\}_{u,v\in[n]}, \{\mathbf{X}_u\}_{u\in n})$ sampled from the model described above, we say that $G \sim \mathrm{CSBM}(n, \mathbb{P}, \mathbf{Q})$ or $G \sim \mathrm{CSBM}(n, \mathbb{P}, \mathbf{B}/n)$, where $\mathbf{B}_{ij}$ is seen as the expected number of neighbours in class j, of a node that is in class i. In a lot of places, we will reduce the model to the simpler case with two symmetric classes C_0 and C_1, in which case, $\mathbb{P} = \{\mathbf{P}_\mp\}$, the distributions of the two classes, and $\mathbf{Q} = (p - q)\mathbf{I}_2 + q\mathbf{J}_2$ where p and q are the intra and inter-class edge probabilities. In this case, I use the notation $\mathrm{CSBM}(n, p, q, \theta)$ where θ represents the parameters of $\mathbf{P}_\mp$. For example, for a binary Gaussian mixture model coupled with a two-block symmetric SBM, we write $\mathrm{CSBM}(n, p, q, \boldsymbol{\mu}, \sigma)$.

2.3 Neural Network Architectures

For the first part of my thesis where I analyze existing GNN architectures, I use the following general family of neural networks where any number of graph convolutions can be placed in any layer. In particular, for a network with L layers, here's the definition:

Architecture 1. Given input data $(\mathbf{A}, \mathbf{X})$ where $\mathbf{A} \in \{0,1\}^{n\times n}$ is the adjacency matrix and $\mathbf{X} \in \mathbb{R}^{n\times d}$ is the node feature matrix, define:

$$\begin{aligned}
&\mathbf{H}^{(0)} = \mathbf{X}, \\
&\left.\begin{aligned} f^{(l)}(\mathbf{X}) &= (\mathbf{D}^{-1}\mathbf{A})^{k_l}\mathbf{H}^{(l-1)}\mathbf{W}^{(l)} + \mathbf{b}^{(l)} \\ \mathbf{H}^{(l)} &= \mathrm{ReLU}(f^{(l)}(\mathbf{X})) \end{aligned}\right\} \text{ for } l \in [L], \\
&\hat{\mathbf{y}} = \varphi(f^{(L)}(\mathbf{X})).
\end{aligned}$$

Here, $\varphi(x) = \frac{1}{1+e^{-x}}$ and ReLU are applied element-wise. The final output of the network is represented by $\hat{\mathbf{y}} = \{\hat{y}_u\}_{i\in[n]}$. Note that $\mathbf{D}^{-1}\mathbf{A}$ is the normalized adjacency matrix and k_l denotes the number of graph convolutions placed in layer l. In particular, for a simple MLP with no graphical information, we have $\mathbf{A} = \mathbf{I}_n$.

For the second part of the thesis, we need the following additional notation and pre-processing. Let $\ell \geq 0$ and $L > 0$ be fixed integers. Let $C \geq 2$ be the number of classes. For given data $(\mathbf{A}, \mathbf{X})$ where $\mathbf{A} \in \mathbb{R}^{n\times n}$ is the adjacency matrix of an unweighted undirected graph, and $\mathbf{X} \in \mathbb{R}^{n\times d}$ is the node feature matrix, we perform a pre-computation on the graph to construct a tensor $\tilde{\mathbf{A}}$ as follows:

$$\tilde{\mathbf{A}}^{(0)} = \mathbf{I}, \qquad \tilde{\mathbf{A}}^{(k)} = f(\mathbf{A}^k) \wedge \left(\neg f\left(\sum_{m=0}^{k-1} \mathbf{A}^m\right)\right) \text{ for } k \in \{1, \dots, \ell\},$$

where $f(M)$ for a matrix M returns the entry-wise flattened matrix with $f(M)_{ij} = \mathbf{1}(M_{ij} > 0)$, and $(\wedge, \neg)$ denote the entry-wise bit-wise operators ('and', 'negation') respectively. Note here that $\tilde{\mathbf{A}}^{(k)}$ is an $n \times n$ binary matrix with $\tilde{\mathbf{A}}^{(k)}_{uv} = 1$ if and only if v is present in the distance k neighbourhood of u but not within the distance $(k-1)$ neighbourhood. The idea behind this pre-processing step is the following: for each node u and each $k \in [\ell]$, we want to divide the radius ℓ neighbourhood of u into ℓ groups of nodes, where each group $k \in [\ell]$ consists of nodes that are within discovered the neighbourhood at each distance from a given node. $\tilde{\mathbf{A}}^{(k)}_{u,:}$ models a non-backtracking walk of length k that considers new nodes in the distance-k neighbourhood that were not discovered.

I now propose the graph neural network architecture for the second part of the thesis.

Architecture 2. Given input data $(\mathbf{A}, \mathbf{X})$ where $\mathbf{A} \in \{0,1\}^{n\times n}$ is the adjacency matrix and $\mathbf{X} \in \mathbb{R}^{n\times d}$ is the node feature matrix, define:

$$\begin{aligned} &\mathbf{H}^{(0)} = \mathbf{X}, && \mathbf{H}^{(l)} = \sigma_l(\mathbf{H}^{(l-1)}\mathbf{W}^{(l)} + \mathbf{1}_n \mathbf{b}^{(l)}) \text{ for } l \in [L], \\ &\mathbf{Q} = \text{sigmoid}(\mathbf{Z}), && \mathbf{M}^{(k)}_{u,i} = \log\langle \mathbf{H}^{(L)}_u, \mathbf{Q}^k_i) \rangle \text{ for } k \in [\ell], u \in [n], i \in [C]. \end{aligned}$$

Then the predicted label is given by $\hat{\mathbf{y}} = \{\hat{y}_u\}_{u\in[n]}$, where

$$\hat{y}_u = \operatorname*{argmax}_{i\in[C]} \left(\mathbf{H}^{(L)}_{u,c} + \sum_{k=1}^{\ell} \tilde{\mathbf{A}}^{(k)}_{u,:} \mathbf{M}^{(k)}_{:,i} \right).$$

Here, $\mathbf{H}^{(L)}$ is viewed as the output of a simple L-layer MLP with $\{\sigma_l\}_{l\in[L]}$ being a set of non-linear activation functions. We have $(\mathbf{W}^{(l)}, \mathbf{b}^{(l)})_{l\in[L]}$ as the learnable parameters of this MLP, with suitable dimensions so that $\mathbf{H}^{(L)} \in \mathbb{R}^{n\times C}$. In addition, we introduce the learnable parameter $\mathbf{Z} \in \mathbb{R}^{C\times C}$ which is used to model edge connectivity among all pairs of classes. The quantity $\tilde{\mathbf{A}}^{(k)}_{u,:}\mathbf{M}^{(k)}_{:,i} = \sum_{v\in[n]} \tilde{\mathbf{A}}^{(k)}_{u,v}\mathbf{M}^{(k)}_{v,i}$ is viewed as the sum of messages $\mathbf{M}^{(k)}_{v,i}$ passed by all distance k neighbours of node u.

2.4 Elementary Results

Let us start by stating some standard definitions and probability tools which will be used throughout this work. The first definition is regarding *sub-Gaussian* random variables. Those random variables are characterized by their tail decay.

Definition 2.4.1 (SubGaussian tails, [Ver18])**.** We say that a random variable $\boldsymbol{z}$ follows *sub-Gaussian* distribution if there are positive constants C, v such that for every $t > 0$

$$\mathbf{Pr}\left[|\boldsymbol{z} - \mathbb{E}[\boldsymbol{z}]| > t\right] \leq C\exp(-vt^2).$$

Equivalently, $\boldsymbol{z}$ is sub-Gaussian if $\mathbb{E}[\exp(a(\boldsymbol{z} - \mathbb{E}[\boldsymbol{z}])^2)] \leq 2$ for some $a > 0$ (this condition is known as ψ_2-condition).

The following lemma discusses the behaviour of the maxima of sub-Gaussian random variables.

Lemma 2.4.1 (Max of subGaussian [RH15]). *Let* $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ *be sub-Gaussian random variables with the same mean and sub-Gaussian parameter* $\tilde{\sigma}^2$. *Then,*

$$\mathbb{E}\left[\max_{i \in [n]} (\boldsymbol{x}_i - \mathbb{E}[\boldsymbol{x}_i])\right] \leq \tilde{\sigma}\sqrt{2\log n}.$$

Moreover, for any $t > 0$

$$\mathbf{Pr}\left[\max_{i \in [n]} (\boldsymbol{x}_i - \mathbb{E}[\boldsymbol{x}_i]) > t\right] \leq 2n \exp\left(-\frac{t^2}{2\tilde{\sigma}^2}\right).$$

Next, we define *Lipschitz* functions and state that the function LeakyRelu, i.e., $f(x) = max(ax, x)$ where a is typically a very small number, is Lipschitz.

Definition 2.4.2 (Lipschitz functions). Let $(\mathcal{X}, d_{\mathcal{X}})$ and $(\mathcal{Y}, d_{\mathcal{Y}})$ be metric spaces. A function $f : \mathcal{X} \to \mathcal{Y}$ is called *L-Lipschitz* if there exists $L \in \mathbb{R}$ such that for every $u, v \in \mathcal{X}$

$$d_{\mathcal{Y}}(f(u), f(v)) \leq L \cdot d_{\mathcal{X}}(u, v).$$

Fact 2.4.2. *LeakyRelu is L-Lipschitz with* $L \leq 1$.

Next are a few concentration inequalities used in this work.

Lemma 2.4.3 (Hoeffding's inequality, Theorem 2.2.6 in [Ver18]). *Let* $X_1, X_2, \ldots, X_n$ *be independent random variables such that* $a_i \leq X_i \leq b_i$ *for each* i. *Define the sample mean as:* $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$. *Then, for any* $t > 0$,

$$\mathbf{Pr}\left[\overline{X} - \mathbb{E}[\overline{X}] \geq t\right] \leq \exp\left(-\frac{2n^2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

Similarly,

$$\mathbf{Pr}\left[\overline{X} - \mathbb{E}[\overline{X}] \leq -t\right] \leq \exp\left(-\frac{2n^2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

Lemma 2.4.4 (Chernoff bound [AS04, Han14, Ver18]). *Let* $\chi_1, \ldots, \chi_n$ *be independent and identically distributed (iid) random variables ranging in* $[0, 1]$, *and let* $p = \mathbb{E}[\chi_1]$. *Then for any* $\epsilon \in (0, 1)$, *it holds that*

$$\mathbf{Pr}\left[\left|\frac{1}{n}\sum_{i \in [n]} \chi_i - p\right| > \epsilon\right] < 2\exp\left(-\frac{\epsilon^2 n}{4}\right),$$

and for any $\gamma \in (0, 2]$*, it holds that*

$$\mathbf{Pr}\left[\left|\frac{1}{n}\sum_{i\in[n]}\chi_i - p\right| > \gamma p\right] < 2\exp\left(-\frac{\gamma^2 pn}{4}\right).$$

The following statement considers the optimal Bayes classifier for data generated by the Gaussian mixture model.

Lemma 2.4.5. *Let* $G = (\mathbf{A}, \mathbf{X}) \sim \mathrm{CSBM}(n, p, q, \boldsymbol{\mu}, \sigma)$*. Then, the optimal Bayes classifier for* $\mathbf{X}$ *is realized by the linear classifier.*

$$h(\boldsymbol{x}) = \begin{cases} 0 & \textit{if } \boldsymbol{x}^T\boldsymbol{\mu} \le 0 \\ 1 & \textit{if } \boldsymbol{x}^T\boldsymbol{\mu} > 0 \end{cases}.$$

Proof. For a given data point $\boldsymbol{x}$ and label $y \in \{0, 1\}$, the Bayes classifier is given by

$$h(\boldsymbol{x}) = \operatorname*{argmax}_{c\in\{0,1\}} \mathbf{Pr}\left[y = c \mid \boldsymbol{x}\right].$$

Note that since the class membership variables $\epsilon_1, \ldots, \epsilon_n \sim \mathrm{Ber}(1/2)$ are independent, we have $\mathbf{Pr}[y = 0] = \frac{1}{2}$ and $\mathbf{Pr}[y = 1] = \frac{1}{2}$. Therefore, by Bayes rule

$$\begin{aligned} &\mathbf{Pr}\left[y = c \mid \boldsymbol{x}\right] \\ &= \frac{\mathbf{Pr}[y = c] \cdot f_{\boldsymbol{x}|y}(\boldsymbol{x} \mid y = c)}{\mathbf{Pr}[y = 0]\, f_{\boldsymbol{x}|y=0}(\boldsymbol{x} \mid y = 0) + \mathbf{Pr}[y = 1]\, f_{\boldsymbol{x}|y=1}(\boldsymbol{x} \mid y = 1)} = \frac{1}{1 + \frac{f_{\boldsymbol{x}|y}(\boldsymbol{x}|y=1-c)}{f_{\boldsymbol{x}|y}(\boldsymbol{x}|y=c)}}. \end{aligned}$$

Assume that $\boldsymbol{x} \in C_0$, we have that $h(\boldsymbol{x}) = 0$ if and only if $\mathbf{Pr}[y = 0 \mid \boldsymbol{x}] \ge 1/2$. Therefore, if we consider class $c = 0$ we need that $\frac{f_{\boldsymbol{x}|y}(\boldsymbol{x}|y=1)}{f_{\boldsymbol{x}|y}(\boldsymbol{x}|y=0)} \le 1$. That is,

$$\frac{f_{\boldsymbol{x}|y}(\boldsymbol{x} \mid y = 1)}{f_{\boldsymbol{x}|y}(\boldsymbol{x} \mid y = 0)} = \frac{\exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{x} - \boldsymbol{\mu}\|_2^2\right)}{\exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{x} + \boldsymbol{\mu}\|_2^2\right)} = \exp\left(-\frac{1}{2\sigma^2}\left(\|\boldsymbol{x} - \boldsymbol{\mu}\|_2^2 - \|\boldsymbol{x} + \boldsymbol{\mu}\|_2^2\right)\right) \le 1,$$

which implies that $\boldsymbol{x}^T\boldsymbol{\mu} \le 0$. Similarly, for label $c = 1$ we get that $\boldsymbol{x}^T\boldsymbol{\mu} > 0$. Hence, the Bayes classifier is given by

$$h(\boldsymbol{x}) = \begin{cases} 0 & \text{if } \boldsymbol{x}^T\boldsymbol{\mu} \le 0 \\ 1 & \text{if } \boldsymbol{x}^T\boldsymbol{\mu} > 0 \end{cases},$$

which is a linear classifier. □

I now note the elementary results that supplement the proofs of the main results.

Proposition 2.4.6 (Degree concentration)**.** *Consider* $G \sim \mathrm{CSBM}(n, \{\mathbf{P}_\mp\}, (p-q)\mathbf{I}_2 + q\mathbf{J}_2)$ *with* $y_u \in \{0,1\}$ *for all* $u \in [n]$ *and the density* $p, q = \Omega(\frac{\log^2 n}{n})$*. Then for any* $c > 0$*, with probability at least* $1 - 2n^{-c}$*, we have for all* $u \in [n]$ *that*

$$\mathbf{deg}(u) = \frac{n}{2}(p+q)(1 \pm o_n(1)), \qquad \frac{1}{\mathbf{deg}(u)} = \frac{2}{n(p+q)}(1 \pm o_n(1)),$$

$$\frac{1}{\mathbf{deg}(u)}\left(\sum_{y_v=1} a_{uv} - \sum_{y_v=0} a_{uv}\right) = (2y_u - 1)\,\mathrm{sgn}(p-q)\gamma(p,q)(1 + o_n(1)),$$

where the error term $o_n(1) = O\left(\sqrt{\frac{c}{\log n}}\right)$.

Proof. Note that $\mathbf{deg}(u)$ is a sum of n Bernoulli random variables, hence, we have by the Chernoff bound [Ver18, Section 2] that

$$\mathbf{Pr}\left[\mathbf{deg}(u) \in \left[\frac{n}{2}(p+q)(1-\delta), \frac{n}{2}(p+q)(1+\delta)\right]^{\mathsf{c}}\right] \le 2\exp(-Cn(p+q)\delta^2),$$

for some $C > 0$. We now choose $\delta = \sqrt{\frac{(c+1)\log n}{Cn(p+q)}}$ for a large constant $c > 0$. Note that since $p, q = \Omega(\log^2 n/n)$, we have that $\delta = O(\sqrt{\frac{c}{\log n}}) = o_n(1)$. Then following a union bound over $u \in [n]$, we obtain that with probability at least $1 - 2n^{-c}$,

$$\begin{aligned} \mathbf{deg}(u) &= \frac{n}{2}(p+q)\left(1 \pm O\Big(\sqrt{\frac{c}{\log n}}\Big)\right) \text{ for all } u \in [n], \\ \frac{1}{\mathbf{deg}(u)} &= \frac{2}{n(p+q)}\left(1 \pm O\Big(\sqrt{\frac{c}{\log n}}\Big)\right) \text{ for all } u \in [n]. \end{aligned}$$

Note that $\frac{1}{\mathbf{deg}(u)}\sum_{y_v=b} a_{uv}$ for any $b \in \{0,1\}$ is a sum of independent Bernoulli random variables. Hence, by a similar argument, we have that with probability at least $1 - 2n^{-c}$,

$$\frac{1}{\mathbf{deg}(u)}\left(\sum_{y_v=1} a_{uv} - \sum_{y_v=0} a_{uv}\right) = (2y_u - 1)\frac{p-q}{p+q}(1 + o_n(1)) \text{ for all } u \in [n],$$

which completes the proof. □

Proposition 2.4.7 (Number of common neighbours)**.** *Assume that the graph density is* $p, q = \Omega(\frac{\log n}{\sqrt{n}})$*. Then for any constant* $c > 0$*, with probability at least* $1 - 2n^{-c}$*,*

$$\begin{aligned} |N_1(u) \cap N_1(v)| &= \frac{n}{2}(p^2 + q^2)(1 \pm o_n(1)) && \textit{for all } y_u = y_v, \\ |N_1(u) \cap N_1(v)| &= npq(1 \pm o_n(1)) && \textit{for all } y_u \neq y_v, \end{aligned}$$

where the error term $o_n(1) = O\left(\sqrt{\frac{c}{\log n}}\right)$*.*

Proof. For any two distinct nodes $u, v \in [n]$ we have that the number of common neighbours of u and v is $|N_1(u) \cap N_1(v)| = \sum_{w \in [n]} a_{uw} a_{vw}$. This is a sum of independent Bernoulli random variables, with mean $\mathbb{E}|N_1(u) \cap N_1(v)| = \frac{n}{2}(p^2 + q^2)$ for $y_u = y_v$ and $\mathbb{E}|N_1(u) \cap N_1(v)| = npq$ for $y_u \neq y_v$. Denote $\mu_{uv} = \mathbb{E}|N_1(u) \cap N_1(v)|$. Therefore, by the Chernoff bound [Ver18, Section 2], we have for a fixed pair of nodes (u, v) that

$$\mathbf{Pr}\left[|N_1(u) \cap N_1(v)| \in [\mu_{uv}(1 - \delta_{uv}), \mu_{uv}(1 + \delta_{uv})]^{\mathsf{c}}\right] \leq 2\exp(-C\mu_{uv}\delta_{uv}^2)$$

for some constant $C > 0$. We now choose $\delta_{uv} = \sqrt{\frac{(c+2)\log n}{C\mu_{uv}}}$ for any large $c > 0$. Note that since $p, q = \Omega(\log n / \sqrt{n})$, we have that $\delta_{uv} = O(\sqrt{\frac{c}{\log n}}) = o_n(1)$. Then following a union bound over all pairs $(u, v) \in [n] \times [n]$, we obtain that with probability at least $1 - 2n^{-c}$, for all pairs of nodes (u, v) we have

$$\begin{aligned} |N_1(u) \cap N_1(v)| &= \frac{n}{2}(p^2 + q^2)(1 \pm o_n(1)) && \text{for all } y_u = y_v, \\ |N_1(u) \cap N_1(v)| &= npq(1 \pm o_n(1)) && \text{for all } y_u \neq y_v. \end{aligned}$$

This completes the proof. □

Fact 2.4.8. *For any* $x \in [0, 1]$*,* $\frac{x}{2} \leq \log(1 + x) \leq x$*.*

Fact 2.4.9. *For any non-negative numbers* u, v, x, y *such that* $x, y \leq 1$*,*

$$xu + yv - \frac{1}{2}(xu + yv)^2 \leq 1 - (1 - x)^u (1 - y)^v \leq xu + yv.$$

Chapter 3

Effects of Graph Convolutions

The first set of results is presented in two parts. In the first part, we analyze the binary CSBM data model and develop the intuition behind the effects of graph convolutions. We characterize the precise change in the threshold for linear separability of the data when a graph convolution is used with this data model. In the second part, we extend the idea to multi-layer networks and work with the more complex data model, the XOR-CSBM. Here, we also analyze the effects of graph convolutions in different combinations across multiple layers of a neural network.

3.1 GNN Architecture

For this part, we shall work with Architecture 1. Let θ denote the set of all learnable parameters of the network, $(\mathbf{W}^{(l)}, \mathbf{b}^{(l)})_{l\in[L]}$. For a dataset $(\mathbf{A}, \mathbf{X}, \mathbf{y})$, denote the binary cross-entropy loss of a multi-layer network with parameters θ by $\ell(\mathbf{A}, \mathbf{X}, \mathbf{y}, \theta) = -\frac{1}{n}\sum_{i\in[n]} y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)$, and the optimization problem as

$$\mathrm{OPT}(\mathbf{A}, \mathbf{X}, \mathbf{y}) = \min_{\theta\in\mathcal{C}} \ \ell(\mathbf{A}, \mathbf{X}, \mathbf{y}, \theta), \tag{3.1}$$

where $\mathcal{C}$ denotes a suitable constraint set for θ. For our analyses, we take the constraint set $\mathcal{C}$ to impose the condition $\left\|\mathbf{W}^{(1)}\right\|_2 \le R$ and $\left\|\mathbf{W}^{(l)}\right\|_2 \le 1$ for all $1 < l \le L$, i.e., the weight parameters of all layers $l > 1$ are normalized, while for $l = 1$, the norm is bounded by some fixed value R. This constraint is necessary for the optimization algorithm to terminate because, without the constraint, the value of the loss function can go arbitrarily close to 0 in the case of perfect classification. Furthermore, the parameter R helps us concisely provide bounds for the loss in the theorems for various regimes by bounding the Lipschitz constant of the learned function. In the rest of this chapter, I use $\ell(\mathbf{X}, \mathbf{y}, \theta)$ to denote $\ell(\mathbf{I}_n, \mathbf{X}, \mathbf{y}, \theta)$.

3.2 Binary CSBM with One-layer Networks

Let $(y_k)_{k\in[n]}$ be iid $\mathrm{Ber}(\frac{1}{2})$ random variables that are latent class labels to be learned. Let $\mathbf{y}$ denote the vector with coordinates y_i. Corresponding to these, consider the adjacency matrix $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{n\times n}$ of a graph drawn from a symmetric stochastic block model with two classes, $C_0 = \{i \in [n] : y_i = 0\}$ and $C_1 = C_0^{\mathsf{c}}$. Additionally, consider a feature matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ (also referred to as *node covariates* in the literature) drawn from a two-component Gaussian mixture model. The distribution of the data conditioned on latent variables $(y_k)_{k\in[n]}$ is given by

$$\mathbf{Pr}(a_{ij} = 1) = \begin{cases} p & y_i = y_j \\ q & \text{otherwise} \end{cases}, \qquad\qquad \mathbf{X}_i \sim \mathcal{N}((2y_i - 1)\boldsymbol{\mu}, \sigma^2 \mathbf{I}_d),$$

where $p, q, \boldsymbol{\mu}, \sigma$ are parameters of the model. A sample drawn from this model is denoted as $(\mathbf{A}, \mathbf{X}) \sim \mathrm{CSBM}(n, p, q, \boldsymbol{\mu}, \sigma)$, where $\mathbf{A} \sim \mathrm{SBM}(n, p, q)$ and $\mathbf{X} \sim \mathrm{GMM}(n, \boldsymbol{\mu}, \sigma)$. Furthermore, we denote the convolved data matrix, i.e. the transformed feature matrix obtained after applying a graph convolution to the original data $\mathbf{X}$ as $\tilde{\mathbf{X}} = \mathbf{D}^{-1}\mathbf{A}\mathbf{X}$. We say that two models $\mathrm{CSBM}(n, p, q, \boldsymbol{\mu}, \sigma)$ and $\mathrm{CSBM}(n', p', q', \boldsymbol{\mu}', \sigma')$ are in the same family if $\boldsymbol{\mu} = \boldsymbol{\mu}'$, $\sigma = \sigma'$ and $\mathrm{sgn}(p - q) = \mathrm{sgn}(p' - q')$, i.e., the underlying Gaussian mixture has the same distribution and the homophilic/heterophilic nature of the underlying graph is the same.

3.2.1 Linear Separability

We find that graph convolutions can dramatically improve the separability of a binary CSBM dataset. In particular, adding the graph structure to a dataset and using the corresponding convolution, i.e., working with $\tilde{\mathbf{X}} = \mathbf{D}^{-1}\mathbf{A}\mathbf{X}$ as opposed to simply $\mathbf{X}$, can make a dataset linearly separable when it was not so previously. More precisely, given a sample $(\mathbf{A}, \mathbf{X}) \sim \mathrm{CSBM}(n, p, q, \boldsymbol{\mu}, \sigma)$, we say that $(\mathbf{X}_i)_{i\in[n]}$ is *linearly separable* if there is some unit vector $\hat{\mathbf{v}}$ such that $\langle \mathbf{X}_i, \hat{\mathbf{v}} \rangle < 0$ for all $i \in C_0$ and $\langle \mathbf{X}_i, \hat{\mathbf{v}} \rangle > 0$ for all $i \in C_1$. We say that $(\tilde{\mathbf{X}}_i)_{i\in[n]}$ is linearly separable if the same holds for $\tilde{\mathbf{X}}$. Let us begin by defining two quantities of interest:

$$\gamma(p,q)=\frac{|p-q|}{p+q},\qquad\qquad \zeta(\boldsymbol{\mu},\sigma)=\frac{\|\boldsymbol{\mu}\|_2}{\sigma}. \tag{3.2}$$

Here, $\gamma(p,q)$ is viewed as the signal in the graph, while $\zeta(\boldsymbol{\mu},\sigma)$ is viewed as the signal in the GMM. In the rest of this section, we will denote $\zeta=\zeta(\boldsymbol{\mu},\sigma)$, making the dependence on $\boldsymbol{\mu}$ and σ implicit. Note that intuitively, a larger gap between the intra-edge and inter-edge probabilities signifies a stronger signal in the graph. When $p=q$, the SBM collapses to the Erdös-Rényi model. Similarly, a larger ratio of the distance between the means of the GMM to the standard deviation signifies a stronger signal in the mixture. Note that our definition of γ is similar to the signal identified by several theoretical works on community detection and GNNs [AS15, Abb18, LS20, WYJ+22].

Theorem 1 (Linear separability). *Let $\boldsymbol{\mu}\in\mathbb{R}^d$ and $\sigma>0$. Then for any data sample $G=(\mathbf{A},\mathbf{X})\sim\mathrm{CSBM}(n,p,q,\boldsymbol{\mu},\sigma)$ where $p,q=\Omega(\log^2 n/n)$, we have the following:*

1. *If $\zeta\le K$ for any constant $K>0$ then $\mathbf{Pr}(\{\mathbf{X}_i\}_{i\in[n]}$ are linearly separable$)=o_n(1)$.*
2. *If $\gamma\zeta=\omega\left(\sqrt{\frac{\log n}{n(p+q)}}\right)$ then $\mathbf{Pr}((\tilde{\mathbf{X}}_i)_{i\in[n]}$ is linearly separable$)=1-o_n(1)$.*
3. *Consider a training dataset $(\mathbf{A}',\mathbf{X}')\sim\mathrm{CSBM}(n,p,q,\boldsymbol{\mu},\sigma)$. Let $\mathbf{w}$ be a linear classifier produced by any arbitrary learning algorithm that takes as input the dataset $(\mathbf{A}',\mathbf{X}')$. Additionally, consider a test dataset $(\mathbf{A},\mathbf{X})\sim\mathrm{CSBM}(n,p,q,\boldsymbol{\mu},\sigma)$ drawn independent of $(\mathbf{A}',\mathbf{X}')$. If the product of the two signals $\gamma\zeta\le\frac{K}{\sqrt{n(p+q)}}$ for a constant $K>0$ and $p,q\le 1-\epsilon$ for any constant $\epsilon\in(0,1)$, then*

$$\mathbf{Pr}((\tilde{\mathbf{X}}_i)_{i\in[n]}\text{ is linearly separable by }\mathbf{w})=o_n(1).$$

This is a statement about the fundamental classification threshold of the data $\mathbf{X}$, and how a graph convolution improves that threshold. The first part of this theorem shows that if we consider a two-component GMM in $\mathbb{R}^d$ with the same variance but different means, then if the signal ζ is $o_n(1)$, that is, the means $\pm\boldsymbol{\mu}$ are $O(\sigma)$ apart, with overwhelming probability it is impossible to linearly separate the data. For the second part we find that the convolved data, $\tilde{\mathbf{X}}=\mathbf{D}^{-1}\mathbf{A}\mathbf{X}$, is linearly separable provided the means are a bit more than $\sigma\sqrt{\frac{\log n}{n(p+q)}}$ apart. In other words, we identify the separability regime in terms of the product of the two signals γ and ζ. Furthermore, on this scale the loss decays exponentially in $R\gamma\zeta$. Consequently, as $n(p+q)/2$ is diverging, this regime contains the regime in which the data $(\mathbf{X}_i)_{i\in[n]}$ is not linearly separable and logistic regression fails at perfect classification. We note here that our arguments show that this bound is essentially sharp (within a small logarithmic factor). Finally the third part of the theorem shows that the convolved data is not linearly separable below the identified threshold.

3.2.2 Generalization

Let us now turn to the related question of generalization. Here, we are interested in the performance of the optimizer of (3.1), call it $\mathbf{w}^*$, on out-of-distribution data. In particular, we are interested in an upper bound on the loss achieved for new data $(\mathbf{A}', \mathbf{X}') \sim \mathrm{CSBM}(n', p', q', \boldsymbol{\mu}, \sigma)$. We find that the graph convolution performs well on any out-of-distribution example. Given that the attributes of the test example are drawn from the same Gaussian mixture as the attributes of the training sample, the graph convolution makes accurate predictions with high probability even when the graph is sampled from a different distribution, i.e., $\mathbf{w}^*$ performs nearly optimally even when the values of n', p', and q' are substantially different from those in the training set.

Theorem 2 (Generalization). *Let $(\mathbf{A}, \mathbf{X}) \sim \mathrm{CSBM}(n, p, q, \boldsymbol{\mu}, \sigma)$ and assume that the product of the signals $\gamma(p, q)\zeta \in \Omega\left(\sqrt{\frac{\log n}{n(p+q)}}\right)$. Let $\mathbf{w}^*$ be the optimizer of (3.1) on the sample $(\mathbf{A}, \mathbf{X})$. Then for any test sample $(\mathbf{A}', \mathbf{X}') \sim \mathrm{CSBM}(n', p', q', \boldsymbol{\mu}, \sigma)$ independent of $(\mathbf{A}, \mathbf{X})$ such that $\mathrm{sgn}(p' - q') = \mathrm{sgn}(p - q)$ and $\gamma(p', q')\zeta \in \Omega\left(\sqrt{\frac{\log n}{n(p+q)}}\right)$, we have for any $c > 0$ that*

$$\mathbf{Pr}(\tilde{\mathbf{X}}' \textit{ is linearly separable by } \mathbf{w}^*) = 1 - O(n^{-c}).$$

Note here that while the result for generalization is stated in terms of the binary-cross entropy, the arguments immediately yield that the number of nodes misclassified by the half-space classifier defined by $\mathbf{w}^*$ must vanish with probability tending to 1.

3.2.3 Proof of Theorem 1

Let us briefly discuss the intuition behind the proof of Theorem 1. To show that the data $(\mathbf{X}_i)_{i\in[n]}$ is not linearly separable, we observe that we can decompose the data in the form $\mathbf{X}_i = (2y_i - 1)\boldsymbol{\mu} + \sigma\mathbf{Z}_i$, where $\mathbf{Z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ are iid. The key observation is that when the distance between the means is $2\|\boldsymbol{\mu}\|_2 = O(\sigma)$ then the intersection of the high probability regions of the two components of the mixture is most of the mass of both so that no plane can separate the high probability regions.

To make this precise, we state an elementary lower bound on the misclassification error for the Gaussian mixture alone, i.e., in the absence of the graph. The proof relies on the fact that we can easily compute the Bayes optimal classifier for the Gaussian mixture model. A Bayes classifier, denoted by $h^*(\boldsymbol{x})$, maximizes the posterior probability of observing a label $y = b$, given the input data $\mathbf{x} = \boldsymbol{x}$. More precisely, $h^*(\boldsymbol{x}) = \operatorname{argmax}_{b\in\{0,1\}} \mathbf{Pr}\left[y = b \mid \mathbf{x} = \boldsymbol{x}\right]$, where $\boldsymbol{x} \in \mathbb{R}^d$ represents a single data point. For a sample drawn from the GMM, we compute the precise fraction of data points misclassified by the Bayes classifier in terms of the GMM signal ζ, and hence, provide a lower bound on the misclassification error for any arbitrary classifier.

Proposition 3.2.1 (Misclassification error in a GMM)**.** *Let* $\{\mathbf{X}_i\}_{i\in[n]}$ *be a dataset sampled from a Gaussian mixture with means* $\pm\boldsymbol{\mu} \in \mathbb{R}^d$ *and variance* $\sigma^2\mathbf{I}_d$*. Let* ζ *be the GMM signal in* (3.2)*. Then for every classifier* $h(\boldsymbol{x}) : \mathbb{R}^d \to \{0, 1\}$*, we have for any constant* $c > 0$ *that with probability at least* $1 - n^{-c}$*, the fraction of data points misclassified by* h *is*

$$M_h(n) \geq \Phi_{\mathsf{c}}(\zeta) - \sqrt{\frac{c\log n}{n}}.$$

Proof. Note that $\mathbf{Pr}\left[y = 0\right] = \mathbf{Pr}\left[y = 1\right] = \frac{1}{2}$. Let $f_{\mathbf{x}}(\boldsymbol{x})$ denote the density function of a continuous random vector $\mathbf{x}$. Therefore, for any $b \in \{0, 1\}$,

$$\mathbf{Pr}\left[y = b \mid \mathbf{x} = \boldsymbol{x}\right] = \frac{\mathbf{Pr}\left[y = b\right] f_{\mathbf{x}|y}(\boldsymbol{x} \mid y = b)}{\sum_{c\in\{0,1\}} \mathbf{Pr}\left[y = c\right] f_{\mathbf{x}|y}(\boldsymbol{x} \mid y = c)} = \frac{1}{1 + \frac{f_{\mathbf{x}|y}(\boldsymbol{x}|y=1-b)}{f_{\mathbf{x}|y}(\boldsymbol{x}|y=b)}}.$$

A direct calculation using the densities gives $\frac{f_{\mathbf{x}|y}(\boldsymbol{x}|y=1)}{f_{\mathbf{x}|y}(\boldsymbol{x}|y=0)} = \exp\left(\frac{2\langle \boldsymbol{x},\boldsymbol{\mu}\rangle}{\sigma^2}\right)$.

The decision regions are identified by: $\mathbf{Pr}\left[y = 0 \mid \mathbf{x}\right] \geq 1/2$ and $\mathbf{Pr}\left[y = 0 \mid \mathbf{x}\right] < 1/2$ for assigning labels 0 and 1 respectively. Thus, we obtain the classifier

$$h^*(\boldsymbol{x}) = \mathbf{1}(\langle \boldsymbol{x}, \hat{\boldsymbol{\mu}}\rangle > 0) = \begin{cases} 0 & \langle \boldsymbol{x}, \hat{\boldsymbol{\mu}}\rangle \leq 0 \\ 1 & \langle \boldsymbol{x}, \hat{\boldsymbol{\mu}}\rangle > 0 \end{cases}. \tag{3.3}$$

We now consider a sample of n data points, $\{\mathbf{X}_i\}_{i\in[n]}$, drawn from the binary GMM with means $\pm\boldsymbol{\mu} \in \mathbb{R}^d$ and variance $\sigma^2\mathbf{I}_d$. For perfect classification by the Bayes classifier, we require for every $i \in [n]$ that $\langle \mathbf{X}_i, \hat{\boldsymbol{\mu}}\rangle \leq 0$ for $y_i = 0$ and $\langle \mathbf{X}_i, \hat{\boldsymbol{\mu}}\rangle > 0$ for $y_i = 1$. Note that the random variable $\mathbf{Y}_i = \langle \mathbf{X}_i, \hat{\boldsymbol{\mu}}\rangle$ is Gaussian with mean $(2y_i - 1)\left\|\boldsymbol{\mu}\right\|_2$ and variance σ^2. Consider the case $i \in C_0$. We have for all $t \in \mathbb{R}$ that $\mathbf{Pr}\left(\frac{\mathbf{Y}_i+\|\boldsymbol{\mu}\|_2}{\sigma} \leq t\right) = \Phi(t)$.

Putting $t = \zeta = \frac{\|\boldsymbol{\mu}\|_2}{\sigma}$, we have that $\mathbf{Pr}\left(\mathbf{Y}_i \le 0\right) = \Phi(\zeta)$. Similarly, for $i \in C_1$, we have that $\mathbf{Pr}\left(\mathbf{Y}_i > 0\right) = \Phi(\zeta)$. We thus obtain that each $\mathbf{X}_i$ is misclassified with probability $\Phi_{\mathsf{c}}(\zeta)$. Next, we consider the complete sample of n data points. Define $M_{h^*}(n)$ to be the fraction of misclassified nodes. Let x_i be the indicator random variable $\mathbf{1}(\mathbf{X}_i$ is misclassified). Then x_i are Bernoulli with mean $\Phi(\zeta)$, and we have that $\mathbb{E}[M_{h^*}(n)] = \frac{1}{n}\sum_{i\in[n]}\mathbb{E}[x_i] = \Phi_{\mathsf{c}}(\zeta)$. Using Hoeffding's inequality [Ver18, Theorem 2.2.6], we have that for any $t > 0$,

$$\mathbf{Pr}\left[M_{h^*}(n) \ge \Phi_{\mathsf{c}}(\zeta) - t\right] \ge 1 - \exp(-nt^2).$$

Choosing $t = \sqrt{c \log n / n}$ for any constant $c > 0$ yields

$$\mathbf{Pr}\left[M_{h^*}(n) \ge \Phi_{\mathsf{c}}(\zeta) - o_n(1)\right] \ge 1 - n^{-c}.$$

Since the Bayes classifier minimizes the risk (expected misclassification fraction) by definition, we have for any arbitrary classifier h that $\mathbb{E}[M_h(n)] \ge \mathbb{E}[M_{h^*}(n)]$. Therefore, any classifier h misclassifies at least $\Phi_{\mathsf{c}}(\zeta)$ fraction of data points with high probability. $\square$

Negative Regime without the Graph

We now prove part one of Theorem 1, i.e., the impossibility of classification of $(\mathbf{X}_i)_{i\in[n]}$ into regions identified by their labels. Note that $\Phi_{\mathsf{c}}(\zeta)$ is a decreasing function. It approaches 0 as $\zeta \to \infty$ and $1/2$ as $\zeta \to 0$. The total number of misclassified data points in a sample of size n is then roughly $n\Phi_{\mathsf{c}}(\zeta) \approx \frac{n}{\zeta}\exp(-\zeta^2/2)$. Thus, when ζ is a constant, we expect a constant fraction of misclassifications, while as soon as $\zeta = \Omega_n(\sqrt{2\log n})$, we expect perfect classification with overwhelming probability. Following this intuition, we note that if $\zeta \le K$ then for any $c > 0$, with probability at least $1 - n^{-c}$, we have that the total number of errors is lower bounded as follows:

$$M(n) \ge n\left(\Phi_{\mathsf{c}}(\zeta) - \sqrt{\frac{c\log n}{n}}\right) \ge n\Phi_{\mathsf{c}}(K)\left(1 - \sqrt{\frac{c\log n}{n\Phi_{\mathsf{c}}(K)^2}}\right) = n\Phi_c(K)(1 - o_n(1)).$$

Therefore, $\{\mathbf{X}_i\}_{i\in[n]}$ are not perfectly classified. This completes the proof.

Positive Regime with Graph Convolution

We now consider the convolved data and prove part two of Theorem 1. Let γ, ζ denote the two signals in (3.2). The key observation here is that after a graph convolution, the convolved data points can be written as

$$\tilde{\mathbf{X}}_i \approx (2y_i - 1)\,\mathrm{sgn}(p-q)\gamma\boldsymbol{\mu} + \frac{\sigma\mathbf{Z}_i}{\sqrt{\mathbf{deg}(i)}}.$$

From this we see that, while the means move closer to each other by a factor of γ, the variance is reduced by a factor of $\mathbb{E}\mathbf{deg}(i) \approx \frac{n}{2}(p+q)$. This lowers the threshold for non-separability by the same factor. Consequently, if the distance between the means is a bit larger than $\frac{\sigma}{\gamma}\sqrt{\frac{2}{n(p+q)}}$, or in other words, if the product of the two signals $\gamma\zeta \in \omega\left(\sqrt{\frac{\log n}{n(p+q)}}\right)$, then with high probability we can separate the data by the hyperplane normal to the direction $\hat{\boldsymbol{\mu}}$ and passing through the origin. More precisely, it suffices to take as ansatz, $\mathbf{w} \propto \hat{\boldsymbol{\mu}}$. To formalize this, we first construct a one-layer network that realizes the Bayes classifier in (3.3).

Proposition 3.2.2. *Consider a one-layer network of the form described in Architecture 1, with parameters* $\mathbf{W}^{(1)} = R\hat{\boldsymbol{\mu}}$ *and* $\mathbf{b}^{(1)} = \mathbf{0}$ *for some* $R \in \mathbb{R}^+$*. Then the defined network realizes the Bayes optimal classifier given in* (3.3) *for the binary* $\mathrm{GMM}(n, \boldsymbol{\mu}, \sigma)$.

Proof. Note that the output of the designed network is $\varphi(\mathbf{X}\mathbf{W}^{(1)})$, which is interpreted as the probability with which the network believes that the input is in the class with label 1. The final prediction for the class label is thus assigned to be 1 if the output is $\geq 1/2$, and 0 otherwise. For each $i \in [n]$, we have that the output of the network on data point i is $\hat{y}_i = \varphi(R\langle \mathbf{X}_i, \hat{\boldsymbol{\mu}}\rangle)$. Thus, we have

$$\mathrm{pred}(\mathbf{X}_i) = \mathbf{1}(R\langle \mathbf{X}_i, \hat{\boldsymbol{\mu}}\rangle > 0) = \mathbf{1}(\langle \mathbf{X}_i, \hat{\boldsymbol{\mu}}\rangle > 0),$$

which matches the Bayes classifier in (3.3). □

We now turn to the convolved data. First, we provide a decomposition of $\tilde{\mathbf{X}}$ which we will use frequently throughout the rest of this section. Note that conditionally on $\mathbf{y}$, we have that $\mathbf{X}_i \sim \mathcal{N}((2y_i - 1)\boldsymbol{\mu}, \sigma^2\mathbf{I})$. Thus, we can write

$$\mathbf{X}_i = (2y_i - 1)\boldsymbol{\mu} + \sigma\mathbf{g}_i, \tag{3.4}$$

where $\mathbf{g}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ are iid copies of a standard normal vector.

Lemma 3.2.3. *Conditionally on* $\mathbf{A}$ *and* $\mathbf{y}$*, we have that for any unit vector* $\mathbf{w}$*, any* $c > 0$*, and some* $C > 0$*, with probability at least* $1 - O(n^{-c})$*, we have for every* $i \in [n]$ *that*

$$\langle \tilde{\mathbf{X}}_i - (2y_i - 1)\,\mathrm{sgn}(p-q)\gamma\boldsymbol{\mu}(1 + o_n(1)), \mathbf{w}\rangle = O\left(\sigma\sqrt{\frac{\log n}{n(p+q)}}\right).$$

Proof. Consider the random variables $\tilde{\mathbf{X}}_i = [\mathbf{D}^{-1}\mathbf{A}\mathbf{X}]_i$. For any fixed i, we define $\mathbf{m}(i)$ to be the mean of $\tilde{\mathbf{X}}_i$ conditional to the adjacency matrix $\mathbf{A}$ and class memberships $(y_j)_{j\in[n]}$,

$$\mathbf{m}(i) = \mathbb{E}[\tilde{\mathbf{X}}_i \mid \mathbf{A}, \mathbf{y}] = \frac{1}{\mathbf{deg}(i)} \sum_{j\in[n]} a_{ij}(2y_j - 1)\boldsymbol{\mu} = \frac{1}{\mathbf{deg}(i)}(|C_1 \cap N_i| - |C_0 \cap N_i|)\boldsymbol{\mu}. \quad (3.5)$$

From (3.4) we can write for any unit vector $\mathbf{w}$ that

$$\langle \tilde{\mathbf{X}}_i, \mathbf{w}\rangle = \frac{1}{\mathbf{deg}(i)} \sum_{j\in[n]} a_{ij}\langle \mathbf{X}_j, \mathbf{w}\rangle = \langle \mathbf{m}(i), \mathbf{w}\rangle + \frac{\sigma}{\mathbf{deg}(i)} \sum_{j\in[n]} a_{ij}\langle \mathbf{g}_j, \mathbf{w}\rangle. \quad (3.6)$$

Note that by the Chernoff bound (see [Ver18, Section 2]), we obtain that class sizes $|C_0|, |C_1| = \frac{n}{2}(1 \pm o(1))$ with probability at least $1 - 1/\mathrm{poly}(n)$. Combining this with Proposition 2.4.6, we have that with probability $1 - Cn^{-c}$ for any $c > 0$,

$$\mathbf{m}(i) = (2y_i - 1)\,\mathrm{sgn}(p - q)\gamma\boldsymbol{\mu}(1 + o_n(1)) \ \forall\, i \in [n]. \quad (3.7)$$

Next, we consider the deviation term $F_i = \frac{\sigma}{\mathbf{deg}(i)} \sum_{j\in[n]} a_{ij}\langle \mathbf{g}_j, \mathbf{w}\rangle$. Note that conditioned on $\mathbf{A}$, we have $F_i \sim \mathcal{N}(0, \frac{\sigma^2}{\mathbf{deg}(i)})$. Then by Gaussian concentration we have

$$\mathbf{Pr}(|F_i| > \delta \mid \mathbf{A}) \le 2\exp(-\delta^2\mathbf{deg}(i)/(2\sigma^2)). \quad (3.8)$$

Define the event $Q = Q(t) = |F_i| \le t\ \forall i \in [n]\}$ and let the event B be where the class sizes concentrate around $n/2$ and Proposition 2.4.6 holds. We have

$$\mathbf{Pr}(Q^{\mathsf{c}}) \le \mathbf{Pr}(B \cap Q^{\mathsf{c}}) + \mathbf{Pr}(B^{\mathsf{c}}) \le 2n\exp\left(-C't^2n(p+q)/\sigma^2\right) + Cn^{-c}$$

for any $c > 0$ and some $C, C' > 0$. Subsequently, we have

$$\mathbf{Pr}(B \cap Q) \ge 1 - \mathbf{Pr}(B^{\mathsf{c}}) - \mathbf{Pr}(Q^{\mathsf{c}}) \ge 1 - Cn^{-c} - 2n\exp\left(-C't^2n(p+q)/\sigma^2\right). \quad (3.9)$$

We now choose $t = \sigma\sqrt{\frac{(c+1)\log n}{C'n(p+q)}}$ to obtain $\mathbf{Pr}(B \cap Q) \ge 1 - (C+2)n^{-c}$. We then observe that on the event $B \cap Q$, we have

$$\left|\langle \tilde{\mathbf{X}}_i - \mathbf{m}(i), \mathbf{w}\rangle\right| = |F_i| = O\left(\sigma\sqrt{\frac{\log n}{n(p+q)}}\right),$$

from which the result follows upon recalling (3.7). □

Next, we show that there exists a $\tilde{\mathbf{w}}$ such that the loss incurred on any sample $(\mathbf{A}, \mathbf{X}) \sim \mathrm{CSBM}(n, p, q, \boldsymbol{\mu}, \sigma)$ is exponentially small with a high probability.

Lemma 3.2.4. *Consider a one-layer neural network with the architecture described in Architecture 1, equipped with one graph convolution, and the parameter* $\mathbf{W}^{(1)} = \tilde{\mathbf{w}} = \frac{R}{\sigma}\operatorname{sgn}(p-q)\hat{\boldsymbol{\mu}}$ *for any* $R > 0$*. Consider a sample* $(\mathbf{A}, \mathbf{X}) \sim \mathrm{CSBM}(n, p, q, \boldsymbol{\mu}, \sigma)$*. In the regime* $\gamma\zeta = \omega\left(\sqrt{\frac{\log n}{n(p+q)}}\right)$ *and for some* $C \in [{}^1\!/_2, 1]$*, we have that with probability at least* $1 - O(n^{-c})$ *for any* $c > 0$*,*

$$\ell(\mathbf{A}, \mathbf{X}, \mathbf{y}, \tilde{\mathbf{w}}) = C \exp\left(-R\gamma\zeta(1 \pm o_n(1))\right).$$

Proof. Consider the loss for a single node i with label y_i,

$$\begin{aligned}\ell_i(\mathbf{A}, \mathbf{X}, \mathbf{y}, \tilde{\mathbf{w}}) &= -y_i \log\left(\varphi(\langle \tilde{\mathbf{X}}_i, \tilde{\mathbf{w}}\rangle)\right) - (1-y_i)\log\left(1 - \varphi(\langle \tilde{\mathbf{X}}_i, \tilde{\mathbf{w}}\rangle)\right) \\ &= \log\left(1 + \exp\left((1-2y_i)(\langle \tilde{\mathbf{X}}_i, \tilde{\mathbf{w}}\rangle)\right)\right).\end{aligned}$$

Using Lemma 3.2.3, it follows that with probability at least $1 - O(n^{-c})$ for any $c > 0$, we have that for all $i \in [n]$,

$$\begin{aligned}\langle \tilde{\mathbf{X}}_i, \tilde{\mathbf{w}}\rangle &= (2y_i - 1)\operatorname{sgn}(p-q)\gamma\langle \boldsymbol{\mu}, \tilde{\mathbf{w}}\rangle(1 + o_n(1)) \pm O\left(\sigma \|\tilde{\mathbf{w}}\| \sqrt{\frac{\log n}{n(p+q)}}\right) \\ &= (2y_i - 1)R\gamma\zeta(1 + o_n(1)) \pm O\left(R\sqrt{\frac{\log n}{n(p+q)}}\right),\end{aligned}$$

where the error terms are uniform in i. In the second equation we put the value of $\tilde{\mathbf{w}}$ from the theorem statement. We now observe that if $\zeta\gamma = \omega\left(\sqrt{\frac{\log n}{n(p+q)}}\right)$, then for all $i \in [n]$ and all $R > 0$ we obtain

$$\langle \tilde{\mathbf{X}}_i, \tilde{\mathbf{w}}\rangle = (2y_i - 1)R\gamma\zeta(1 \pm o_n(1)) = (2y_i - 1)CR\gamma\zeta \tag{3.10}$$

for some constant $C > 0$. Hence, with the same probability, the loss incurred for each node $i \in [n]$ is $\ell_i(\mathbf{A}, \mathbf{X}, \mathbf{y}, \tilde{\mathbf{w}}) = \log\left(1 + \exp\left(-CR\gamma\zeta\right)\right)$. Thus, the total loss is given by

$$\ell(\mathbf{A}, \mathbf{X}, \mathbf{y}, \tilde{\mathbf{w}}) = \frac{1}{n}\sum_{i \in [n]} \ell_i(\mathbf{A}, \mathbf{X}, \mathbf{y}, \tilde{\mathbf{w}}) = \log\left(1 + \exp\left(-CR\gamma\zeta\right)\right).$$

The result then follows using Fact 2.4.8. □

To finish the proof of Theorem 1 part two, note that from (3.10), we have that in the said regime, with probability at least $1 - O(n^{-c})$ for any $c > 0$, the quantities $\langle \tilde{\mathbf{X}}_i, \tilde{\mathbf{w}} \rangle$ have the correct signs, and therefore, a network with a graph convolution that outputs $\varphi(\langle \tilde{\mathbf{X}}_i, \tilde{\mathbf{w}} \rangle)$ obtains perfect classification with overwhelming probability. The exponentially decaying expression for the cross-entropy is then obtained directly from Lemma 3.2.4.

Negative regime with Graph Convolution

We now provide the proof of part three of Theorem 1, i.e., the non-separability threshold for the convolved data $\tilde{\mathbf{X}}$. The way we define separability here implicitly includes the desired property of generalization. In particular, consider a training dataset $(\mathbf{A}', \mathbf{X}') \sim \text{CSBM}(n, p, q, \boldsymbol{\mu}, \sigma)$ and a test dataset $(\mathbf{A}, \mathbf{X}) \sim \text{CSBM}(n, p, q, \boldsymbol{\mu}, \sigma)$ drawn independently from $(\mathbf{A}', \mathbf{X}')$ but from the same CSBM. Next, consider an arbitrary learning algorithm $\mathcal{L}(\mathbf{A}', \mathbf{X}')$ that takes as input the training dataset $(\mathbf{A}', \mathbf{X}')$ and outputs a linear classifier $\mathbf{w}$. We will bound the probability that $\mathbf{w}$ is able to perfectly classify the test data $(\mathbf{A}, \mathbf{X})$.

Recall that with probability at least $1 - O(n^{-c})$ with any choice of $c > 0$, the value of $\mathbf{m}(i)$ is given by (3.7). Note that for successful classification, we require $\langle \tilde{\mathbf{X}}_i, \mathbf{w} \rangle < 0$ for $i \in C_0$ and $\langle \tilde{\mathbf{X}}_i, \mathbf{w} \rangle > 0$ for $i \in C_1$. These conditions are equivalent to the event that

$$\gamma \langle \mathbf{w}, \boldsymbol{\mu} \rangle (1 \pm o_n(1)) + \max_{i \in C_0} \frac{\sigma}{\mathbf{deg}(i)} \sum_{j \in [n]} a_{ij} \langle \mathbf{g}_j, \mathbf{w} \rangle < 0, \tag{3.11}$$

$$-\gamma \langle \mathbf{w}, \boldsymbol{\mu} \rangle (1 \pm o_n(1)) + \min_{i \in C_1} \frac{\sigma}{\mathbf{deg}(i)} \sum_{j \in [n]} a_{ij} \langle \mathbf{g}_j, \mathbf{w} \rangle > 0, \tag{3.12}$$

where the error term $o_n(1) = O(\frac{1}{\sqrt{\log n}})$. Denote $\Delta = \mathbb{E} \deg = \frac{n}{2}(p+q)$ and $T = C_0$ if $|C_0| \le |C_1|$, $T = C_1$ otherwise. Then we can bound the probability of the above event by the following probability.

$$\mathbf{Pr}\left(\max_{i \in T} \frac{\sigma}{\mathbf{deg}(i)} \sum_{j \in [n]} a_{ij} \langle \mathbf{g}_j, \mathbf{w} \rangle \le -\gamma \langle \mathbf{w}, \boldsymbol{\mu} \rangle \right) (1 \pm o_n(1)) \tag{3.13}$$

$$\le \mathbf{Pr}\left(\max_{i \in T} \frac{\sigma}{\mathbf{deg}(i)} \sum_{j \in [n]} a_{ij} \langle \mathbf{g}_j, \mathbf{w} \rangle \le \gamma \left\| \boldsymbol{\mu} \right\|_2 (1 \pm o_n(1)) \right) \quad \text{using Cauchy-Schwarz} \tag{3.14}$$

$$\le \mathbf{Pr}\left(\max_{i \in T} \frac{\sigma}{\mathbf{deg}(i)} \sum_{j \in [n]} a_{ij} \langle \mathbf{g}_j, \mathbf{w} \rangle \le \gamma \zeta (1 \pm o_n(1)) \right) \quad \text{using } \zeta = \left\| \boldsymbol{\mu} \right\|_2 / \sigma \tag{3.15}$$

$$\leq \mathbf{Pr}\left(\max_{i\in T}\langle \mathbf{Z}_i, \mathbf{w}\rangle \leq \frac{K}{\sqrt{\Delta}}(1 \pm o_n(1))\right) \qquad \text{using } \gamma\zeta \leq \frac{K}{\sqrt{\Delta}}, \tag{3.16}$$

where the random vectors $\mathbf{Z}_i = \frac{1}{\mathbf{deg}(i)}\sum_{j\in[n]} a_{ij}\mathbf{g}_j \sim \mathcal{N}(\mathbf{0}, \frac{1}{\mathbf{deg}(i)}\mathbf{I}_d)$. We will now utilize Sudakov's minoration inequality [Ver18, Section 7.4] to obtain a lower bound on the expected supremum of the random process $\{\langle \mathbf{Z}_i, \mathbf{w}\rangle\}_{i\in T}$, and then use Borell's inequality [AT07, Section 2.1] to upper bound the probability in (3.16).

Denote the set $J_{ij} = (N_i \cup N_j) \setminus (N_i \cap N_j)$, and note that

$$\langle \mathbf{Z}_i, \mathbf{w}\rangle - \langle \mathbf{Z}_j, \mathbf{w}\rangle = \frac{(1 \pm o_n(1))}{\Delta}\sum_{l\in J_{ij}}\langle \mathbf{g}_l, \mathbf{w}\rangle.$$

To apply Sudakov's minoration result, we also define the canonical metric over the index set T for any $i, j \in T$:

$$d_T(i,j) = \sqrt{\mathbb{E}[(\langle \mathbf{Z}_i, \mathbf{w}\rangle - \langle \mathbf{Z}_j, \mathbf{w}\rangle)^2]} = \frac{\sqrt{|J_{ij}|}}{\Delta}(1 \pm o_n(1)). \tag{3.17}$$

For any $i, j \in T$ with $i \neq j$ and a node l, the probability of l being a neighbor of exactly one of i, j is $2p(1-p)$ if $l \in T$ and $2q(1-q)$ if $l \in [n]\setminus T$. Thus, $|J_{ij}|$ is a sum of independent Bernoulli random variables and $\mathbb{E}|J_{ij}| = n(p(1-p)+q(1-q))$. Hence, by the multiplicative Chernoff bound we obtain that for any $\delta \in (0,1)$,

$$\mathbf{Pr}(||J_{ij}| - \mathbb{E}|J_{ij}|| > \delta\mathbb{E}|J_{ij}|) \leq 2\exp\left(-\frac{\delta^2\mathbb{E}|J_{ij}|}{3}\right).$$

Since $p, q = \omega(\frac{\log^2 n}{n})$, we have that $\mathbb{E}|J_{ij}| = n(p(1-p)+q(1-q)) = \omega(\log^2 n)$. Therefore, choosing $\delta = \frac{\sqrt{C\log n}}{\mathbb{E}|J_{ij}|}$ for any large constant C, we obtain that with probability at least $1 - O(n^{-c})$, $|J_{ij}| \geq n(p+q-p^2-q^2)(1-\delta) = n(p+q-p^2-q^2)(1-o_n(1)) = \Omega(\Delta)$ since $p, q \leq 1-\epsilon$ for a fixed $\epsilon > 0$. Therefore, we have that

$$d_T(i,j) = \frac{\sqrt{|J_{ij}|}}{\Delta}(1 \pm o_n(1)) = \Omega\left(\frac{1}{\sqrt{\Delta}}\right).$$

Under the above mentioned event, let $\epsilon_0 = \min_{i,j\in S} d_T(i,j)$. Then for an ϵ_0-covering of the set, we need every point of the set, i.e., $N(T, d_T, \epsilon_0) = |T|$. Putting this information in Sudakov's minoration inequality, we obtain that $\mathbb{E}[\max_i\langle \mathbf{Z}_i, \mathbf{w}\rangle] \geq C\epsilon_0\sqrt{\log n}$. Since $\epsilon_0 = \Omega(1/\sqrt{\Delta})$, this gives us that for some suitable $C > 0$,

$$\mathbb{E}[\max_i\langle \mathbf{Z}_i, \mathbf{w}\rangle] \geq C\sqrt{\frac{\log n}{\Delta}}. \tag{3.18}$$

We now use Borell's inequality [AT07, Section 2.1] so that for any $t > 0$,

$$\begin{aligned} &\mathbf{Pr}\left(\max_{i\in C_0}\langle \mathbf{Z}_i, \mathbf{w}\rangle \le \mathbb{E}[\max_{i\in C_0}\langle \mathbf{Z}_i, \mathbf{w}\rangle] - t\right) \le 2\exp(-t^2\mathbf{deg}(i)) \\ \implies \quad &\mathbf{Pr}\left(\max_{i\in C_0}\langle \mathbf{Z}_i, \mathbf{w}\rangle \le c\sqrt{\frac{\log n}{\Delta}} - t\right) \le 2\exp(-t^2\mathbf{deg}(i)) \qquad \text{using (3.18)}. \end{aligned}$$

Choose $t = c\sqrt{\frac{\log n}{\Delta}} - \frac{K}{\sqrt{\Delta}} = \Omega\left(\sqrt{\frac{\log n}{\Delta}}\right)$ and combine with the event for class-size and degree concentration from Proposition 2.4.6, so that for some constant $c' > 0$,

$$\mathbf{Pr}\left(\max_{i\in T}\langle \mathbf{Z}_i, \mathbf{w}\rangle \le \frac{K}{\sqrt{\Delta}}\right) \le 2n^{-c} = o_n(1). \tag{3.19}$$

Thus, for any such $\mathbf{w}$, we showed that the probability that $\mathbf{w}$ perfectly classifies the convolved data $(\mathbf{A}, \mathbf{X}) \sim \mathrm{CSBM}(n, p, q, \boldsymbol{\mu}, \sigma)$ is $o_n(1)$. This completes the proof.

3.2.4 Proof of Theorem 2

Let us now prove Theorem 2 by characterizing $\mathbf{w}^*$. The key point for this result is that the optimizer of (3.1), $\mathbf{w}^*$, must be close to the ansatz we construct, and should separate the data better than the ansatz. Secondly, we observe that the chosen ansatz does not depend on the particular values of p and q, but only the sign of $p - q$. As such, it can be shown that $\tilde{\mathbf{w}}$ performs well on out-of-distribution data corresponding to different values of p', q' with $\mathrm{sgn}(p' - q') = \mathrm{sgn}(p - q)$. Combining these two observations then shows that $\mathbf{w}^*$ also performs well on the out-of-distribution data. Define the following quantities.

$$\mathbf{m}_0 = -\,\mathrm{sgn}(p - q)\gamma(p, q)\boldsymbol{\mu}, \qquad \mathbf{m}_1 = \mathrm{sgn}(p - q)\gamma(p, q)\boldsymbol{\mu}. \tag{3.20}$$

We then have the following Lemma about the expected behaviour of $\mathbf{w}^*$.

Lemma 3.2.5. *For any $R > 0$, let $\mathbf{w}^*$ be the optimizer to the problem in (3.1) for a given training sample $(A, X) \sim \mathrm{CSBM}(n, p, q, \boldsymbol{\mu}, \sigma)$ with $\boldsymbol{\mu} \in \mathbb{R}^d$ and with norm constraint R. Consider the regime where $\gamma\zeta = \Omega\left(\sqrt{\frac{\log n}{n(p+q)}}\right)$. Then for any $c > 0$ fixed but large enough, with probability at least $1 - O(n^{-c})$ we have that for any $R > 0$,*

$$\mathbf{w}^* = R\,\mathrm{sgn}(p - q)\hat{\boldsymbol{\mu}}(1 - o_n(1)). \tag{3.21}$$

Furthermore, we have that

$$\langle \mathbf{m}_0, \mathbf{w}^*\rangle \le -R\sigma\gamma(p, q)\zeta(1 \pm o_n(1)), \qquad \langle \mathbf{m}_1, \mathbf{w}^*\rangle \ge R\sigma\gamma(p, q)\zeta(1 \pm o_n(1)). \tag{3.22}$$

Proof. Fix $R > 0$ and let $\mathbf{w}^*$ be the solution to the problem in (3.1) with norm constraint R. Let the training sample be $(\mathbf{A}, \mathbf{X}) \sim \text{CSBM}(n, p, q, \boldsymbol{\mu}, \sigma)$. Then we have that

$$\ell(\mathbf{A}, \mathbf{X}, \mathbf{y}, \mathbf{w}^*) \leq \ell(\mathbf{A}, \mathbf{X}, \mathbf{y}, \tilde{\mathbf{w}}),$$

where $\tilde{\mathbf{w}}$ is defined in Lemma 3.2.4. Let $\tilde{\mathbf{X}}_i = (\mathbf{D}^{-1}\mathbf{A}\mathbf{X})_i$. Now we focus our scope to the event that for every $i \in [n]$ and $R > 0$, we have $\langle \tilde{\mathbf{X}}_i, \tilde{\mathbf{w}} \rangle = (2y_i - 1)R\gamma(p, q)\zeta(1 + o_n(1))$. Note that from (3.10), this event occurs with probability at least $1 - O(n^{-c})$ for c large but constant. Since $\mathbf{w}^*$ is a solution to (3.1) and $\|\tilde{\mathbf{w}}\|_2 = R/\sigma$, on this event we have for all i that

$$(2y_i - 1)\langle \tilde{\mathbf{X}}_i, \mathbf{w}^* \rangle \geq R\sigma\gamma(p, q)\zeta(1 \pm o_n(1)).$$

Now Lemma 3.2.3 implies that with probability at least $1 - O(n^{-c})$, for all i we also have

$$\left| \langle \tilde{\mathbf{X}}_i - \mathbf{m}_{y_i}(1 + o_n(1)), \mathbf{w}^* \rangle \right| = O\left(\|\mathbf{w}^*\| \, \sigma \sqrt{\frac{\log n}{n(p+q)}} \right)$$

Since $\|\mathbf{w}^*\|_2 \leq R$, we conclude that $(1 - 2y_i)\langle \mathbf{m}_{y_i}, \mathbf{w}^* \rangle \leq -R\sigma\gamma(p, q)\zeta(1 \pm o_n(1))$, that is, (3.22) holds as desired. Now subtracting the two inequalities in (3.22) we obtain that

$$\langle \mathbf{m}_1 - \mathbf{m}_0, \mathbf{w}^* \rangle = 2 \, \text{sgn}(p - q)\gamma(p, q)\langle \boldsymbol{\mu}, \mathbf{w}^* \rangle \geq 2R\sigma\gamma(p, q)\zeta(1 \pm o_n(1)). \tag{3.23}$$

This implies that $\|\mathbf{w}^*\|_2 \geq R(1 - o_n(1))$. Combined with the fact that $\|\mathbf{w}^*\| \leq R$ due to the optimization constraint in (3.1), we have

$$1 - o_n(1) \leq \frac{\langle \boldsymbol{\mu}, \mathbf{w}^* \rangle}{\|\boldsymbol{\mu}\|_2 \, \|\mathbf{w}^*\|_2} \leq 1.$$

Thus, the solution of the optimization problem is the suitably scaled vector $\hat{\boldsymbol{\mu}}$. □

We now consider a test sample $(\mathbf{A}', \mathbf{X}') \sim \text{CSBM}(n', p', q', \boldsymbol{\mu}, \sigma)$. Let $\tilde{\mathbf{X}}'$ be the corresponding convolution $\mathbf{D}'^{-1}\mathbf{A}'\mathbf{X}'$. Similar to (3.7) and (3.20) we also define $\mathbf{m}'(i)$, $\mathbf{m}'_0$ and $\mathbf{m}'_1$ corresponding to the sample $(\mathbf{A}', \mathbf{X}')$. We restrict our calculations to the case where $y_i = 0$, since the argument for $y_i = 1$ is similar. Note that

$$\mathbf{m}'_0 - \mathbf{m}_0 = \frac{2(pq' - qp')}{(p+q)(p'+q')}\boldsymbol{\mu}.$$

From Lemmas 3.2.3 and 3.2.5, we see that for $c' > 0$ large but $O(1)$, with probability at least $1 - O((n')^{-c'})$ we have that for any $R > 0$

$$\langle \tilde{\mathbf{X}}'_i, \mathbf{w}^* \rangle = \langle \mathbf{m}'_{y_i}, \mathbf{w}^* \rangle (1 + o_n(1)) \; \forall \, i \in [n'].$$

Therefore, by the same lemmas, we have that for any $c, c' > 0$ large enough, with probability $1 - O((n')^{-c'} + n^{-c})$, when $y_i = 0$

$$\begin{aligned}
\langle \tilde{\mathbf{X}}'_i, \mathbf{w}^* \rangle &= \langle \mathbf{m}'_0, \mathbf{w}^* \rangle (1 + o_n(1)) \\
&= \langle \mathbf{m}'_0 - \mathbf{m}_0, \mathbf{w}^* \rangle (1 + o_n(1)) + \langle \mathbf{m}_0, \mathbf{w}^* \rangle (1 + o_n(1)) \\
&\leq \frac{2(pq' - qp')}{(p+q)(p'+q')} \langle \boldsymbol{\mu}, \mathbf{w}^* \rangle - R\sigma\gamma(p,q)\zeta(1 \pm o_n(1)) \\
&\leq \frac{2R\,\mathrm{sgn}(p-q)\,\|\boldsymbol{\mu}\|_2\,(pq' - qp')}{(p+q)(p'+q')} - R\sigma\gamma(p,q)\zeta(1 \pm o_n(1)) \\
&= R\,\|\boldsymbol{\mu}\|_2\,\mathrm{sgn}(p-q) \left(\frac{2(pq' - qp')}{(p+q)(p'+q')} - \frac{p-q}{p+q} \right) (1 - o_n(1)) \\
&= -R\,\|\boldsymbol{\mu}\|_2\,\mathrm{sgn}(p-q)\,\mathrm{sgn}(p'-q')\gamma(p',q')(1 - o_n(1)) \\
&= -R\,\|\boldsymbol{\mu}\|_2\,\gamma(p',q')(1 - o_n(1)).
\end{aligned}$$

The first inequality above uses (3.22), while the second inequality follows from Lemma 3.2.5. In the last equation, we used $\mathrm{sgn}(p-q) = \mathrm{sgn}(p'-q')$. Similarly, for $y_i = 1$ we obtain

$$\langle \tilde{\mathbf{X}}'_i, \mathbf{w}^* \rangle \geq R\,\|\boldsymbol{\mu}\|_2\,\gamma(p',q')(1 - o_n(1)).$$

This shows that the classifier $\mathbf{w}^*$ obtains correct signs for $\tilde{\mathbf{X}}'$, i.e., negative for $i \in C_0$ and positive for $i \in C_1$, implying perfect classification of all the nodes.

3.3 XOR-CSBM with Multi-layer Networks

In this section, we look at an XOR arrangement of CSBM, which we refer to as the XOR-CSBM. The choice of this data model is inspired by the fact that it is not linearly separable but is non-linearly separable. Therefore, a single-layer network fails to classify the data from this model and a multi-layer network is required. We have two classes here as well, C_0 and C_1, and the graph $\mathbf{A}$ is sampled from a symmetric two-block SBM as before. However, the GMM now has four components instead of two. We also have latent variables $\eta_i \sim \mathrm{Ber}(1/2)$ for the cluster membership. More precisely, let $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ be fixed vectors in $\mathbb{R}^d$, such that $\|\boldsymbol{\mu}\|_2 = \|\boldsymbol{\nu}\|_2$ and $\langle \boldsymbol{\mu}, \boldsymbol{\nu} \rangle = 0$. The distribution of the data $(\mathbf{A}, \mathbf{X})$ conditioned on latent variables $(y_k)_{k \in [n]}$ and $(\eta_k)_{k \in [n]}$ is given by

$$\mathbf{Pr}(a_{ij} = 1) = \begin{cases} p & y_i = y_j \\ q & \text{otherwise} \end{cases}, \qquad \mathbf{X}_i \sim \mathcal{N}((2\eta_i - 1)((1 - y_i)\boldsymbol{\mu} + y_i\boldsymbol{\nu}), \sigma^2 \mathbf{I}_d),$$

For data $(\mathbf{A}, \mathbf{X}) = (\{a_{ij}\}_{i,j\in[n]}, \{\mathbf{X}_i\}_{i\in n})$ sampled from this model, we say that $(\mathbf{A}, \mathbf{X}) \sim$ XOR-CSBM$(n, p, q, \boldsymbol{\mu}, \boldsymbol{\nu}, \sigma)$, where $\mathbf{A} \sim$ SBM(n, p, q) and $\mathbf{X} \sim$ XOR-GMM$(n, \boldsymbol{\mu}, \boldsymbol{\nu}, \sigma)$.

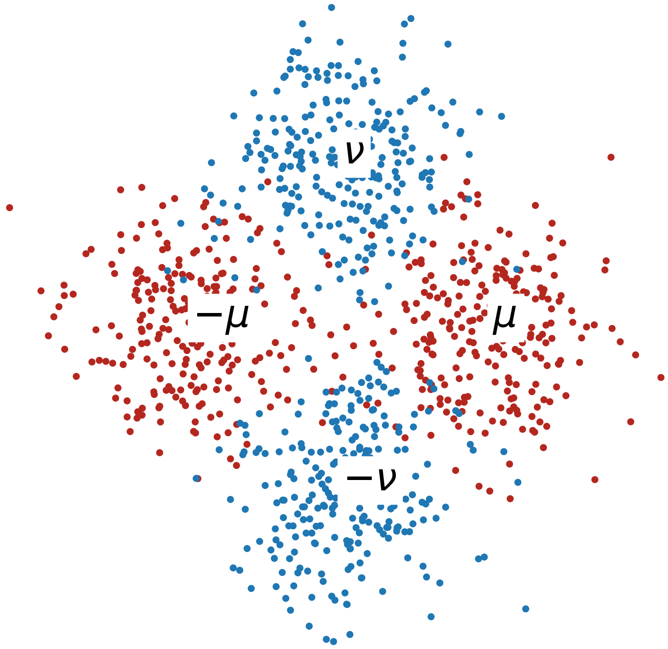


Figure 3.1: Example of a 2D Gaussian mixture with orthogonal means $\pm\boldsymbol{\mu}$ and $\pm\boldsymbol{\nu}$.

3.3.1 Setting up the Baseline

Before stating the main result about the benefits and performance of graph convolutions, we set up a comparative baseline in the setting where graphical information is absent. To this end, we completely characterize the classification threshold for the XOR-GMM data model in terms of the distance between the means of the mixture.

Theorem 3 (Misclassification error)**.** *Let* $\mathbf{X} \sim$ *XOR-GMM*$(n, \boldsymbol{\mu}, \boldsymbol{\nu}, \sigma)$ *be sampled from the XOR Gaussian mixture. Then the following hold:*

1. *Assume that* $\|\boldsymbol{\mu} - \boldsymbol{\nu}\|_2 \leq K\sigma$ *and let* $h(\mathbf{x}) : \mathbb{R}^d \to \{0, 1\}$ *be any binary classifier. Then for any* $K > 0$ *and any* $\epsilon \in (0, 1)$*, at least a fraction* $2\Phi_{\mathrm{c}}\left({}^{K}\!/_{2}\right)^2 - O(n^{-\epsilon/2})$ *of all data points are misclassified by* h *with probability at least* $1 - \exp(-2n^{1-\epsilon})$*.*

2. *For any $\epsilon > 0$, if the distance between the means is $\|\boldsymbol{\mu} - \boldsymbol{\nu}\|_2 = \Omega(\sigma(\log n)^{\frac{1}{2}+\epsilon})$, then there exist a two-layer and a three-layer network that perfectly classify the data with probability at least $1 - O(n^{-c})$ for any $c > 0$.*

Part one of Theorem 3 shows that if the means of the features of the two classes are at most $O(\sigma)$ apart then with overwhelming probability, there is a constant fraction of points that are misclassified. Note that the fraction of misclassified points is $2\Phi_{\mathrm{c}}(K/2)^2$, which approaches 0 as $K \to \infty$ and approaches $1/2$ as $K \to 0$, signifying that if the means are very far apart then we successfully classify all data points, while if they coincide then we always misclassify roughly half of all data points. Furthermore, note that if $K = c\sqrt{\log n}$ for some constant $c \in [0, 1)$, then the total number of points misclassified is $2n\Phi_{\mathrm{c}}(K)^2 \asymp \frac{n}{K^2}e^{-K^2} \asymp \frac{n^{1-c^2}}{\log n} = \Omega(1)$. Thus, intuitively, $K \asymp \sqrt{\log n}$ is the threshold beyond which learning methods are expected to perfectly classify the data. This is formalized in part two of the theorem, which shows that if the means are roughly $\omega(\sigma\sqrt{\log n})$ apart then the data can be perfectly classified.

3.3.2 Improvement through Graph Convolutions

We now state the results that explain the effects of graph convolutions in multi-layer networks with the architecture described in Architecture 1. Denote $\beta = \|\boldsymbol{\mu} - \boldsymbol{\nu}\|_2$ and $\beta' = \beta/\sqrt{2} = \|\boldsymbol{\mu}\|_2 = \|\boldsymbol{\nu}\|_2$. Let $\mathrm{erf}(t) = 2\Phi(t\sqrt{2}) - 1$ be the Gauss error function and $\psi(t) = t\,\mathrm{erf}(t) - (1 - \exp(-t^2))/\sqrt{\pi}$.

Let us begin by defining the signals from the two sources of information (the graph $\mathbf{A}$ and the features $\mathbf{X}$) in the XOR-CSBM model, given by

$$\gamma = \frac{|p-q|}{p+q}, \qquad \zeta = \psi\left(\frac{\beta}{2\sigma}\right). \tag{3.24}$$

We now state the most important result of our work.

Theorem 4. *Let $(\mathbf{A}, \mathbf{X}) \sim$ XOR-CSBM$(n, p, q, \boldsymbol{\mu}, \boldsymbol{\nu}, \sigma)$. Let γ and ζ be the signals defined in (3.24). Then there exist a two-layer and a three-layer network such that:*

- *If the intra-class and inter-class edge probabilities are $p, q = \Omega(\frac{\log^2 n}{n})$, and it holds that $\gamma\zeta = \omega\left(\sqrt{\frac{\log n}{n(p+q)}}\right)$, then for any $c > 0$, with probability at least $1 - O(n^{-c})$, the networks equipped with a graph convolution in the second or the third layer perfectly classify all the nodes.*

- *If* $p, q = \Omega(\frac{\log n}{\sqrt{n}})$ *and* $\gamma^2\zeta = \omega\left(\sqrt{\frac{\log n}{n}}\right)$*, then for any* $c > 0$*, with probability at least* $1 - O(n^{-c})$*, the networks with any combination of two graph convolutions in the second and/or the third layers perfectly classify all the nodes.*

To understand Theorem 4 intuitively, it helps to consider the case where $\gamma(p, q) = \Omega(1)$, i.e., when the graph has a non-vanishing signal. Part one of the theorem then states that under the assumption that $p, q = \Omega(\log^2 n/n)$, a single graph convolution improves the classification threshold for $\|\boldsymbol{\mu} - \boldsymbol{\nu}\|_2$, the distance between the means by a factor of at least $1/\sqrt[4]{n(p+q)}$ and up to $1/\sqrt{n(p+q)}$ as compared to the case without the graph. Similarly, part two shows that with a slightly stronger assumption on the graph density, we observe further improvement in the threshold up to a factor of at least $1/\sqrt[4]{n}$ and up to $1/\sqrt{n}$.

Note that although the regime of graph density is different for part two of the theorem, the result itself is an improvement. In particular, if $p, q = \Omega(\log n/\sqrt{n})$ then part one of the theorem states that one graph convolution achieves an improvement of at least $1/\sqrt[8]{n}$, while part two states that two convolutions improve it to at least $1/\sqrt[4]{n}$. However, we also emphasize that in the regime where the graph is dense, i.e., when $p, q = \Omega_n(1)$, two graph convolutions do not have a significant advantage over one graph convolution. Our experiments in Section 3.5.1 demonstrate this effect.

We also note that an artifact of the XOR-CSBM data model is that a graph convolution in the first layer hurts the classification accuracy. Hence, we only consider networks with no convolutions in the first layer, i.e., $k_1 = 0$.

3.3.3 Placement of Graph Convolutions

We observe that the improvements in the classification capability of a multi-layer network depend on the number of convolutions, and do not depend on where the convolutions are placed. In particular, for the XOR-CSBM data model, putting the same number of convolutions among the second and/or the third layer in any combination achieves mutually similar improvements in the classification task.

Corollary 4.1. *Consider the data model XOR-CSBM*$(n, p, q, \boldsymbol{\mu}, \boldsymbol{\nu}, \sigma)$ *and the network architecture described in Architecture 1. Then we have the following:*

- *Assume that* $p, q = \Omega(\log^2 n/n)$*, and consider the three-layer network characterized by part one of Theorem 4, with one graph convolution. For this network, placing the graph convolution in the second layer* ($k_2 = 1, k_3 = 0$) *obtains the same results as placing it in the third layer* ($k_2 = 0, k_3 = 1$)*.*

- *Assume that* $p, q = \Omega({}^{\log n}/_{\sqrt{n}})$*, and consider the three-layer network characterized by part two of Theorem 4, with two graph convolutions. For this network, placing both convolutions in the second layer* ($k_2 = 2, k_3 = 0$) *or both of them in the third layer* ($k_2 = 0, k_3 = 2$) *obtains the same results as placing one convolution in the second layer and one in the third layer* ($k_2 = 1, k_3 = 1$)*.*

Corollary 4.1 is immediate from Theorem 4. In Section 3.5, we also show extensive experiments on both synthetic and real-world data that demonstrate this result.

3.3.4 Proof of Theorem 3

We begin by computing the Bayes optimal classifier for the XOR-GMM similar to that in Proposition 3.2.1.

Lemma 3.3.1. *For some fixed* $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^d$ *and* $\sigma > 0$*, the Bayes optimal classifier,* $h^*(\boldsymbol{x}) : \mathbb{R}^d \to \{0, 1\}$ *for the data model XOR-GMM*$(n, \boldsymbol{\mu}, \boldsymbol{\nu}, \sigma)$ *is given by*

$$h^*(\boldsymbol{x}) = \mathbf{1}(|\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle| < |\langle \boldsymbol{x}, \boldsymbol{\nu}\rangle|) = \begin{cases} 0 & |\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle| \geq |\langle \boldsymbol{x}, \boldsymbol{\nu}\rangle| \\ 1 & |\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle| < |\langle \boldsymbol{x}, \boldsymbol{\nu}\rangle| \end{cases},$$

where $\mathbf{1}$ *is the indicator function.*

Proof. Note that $\mathbf{Pr}\left[y = 0\right] = \mathbf{Pr}\left[y = 1\right] = \frac{1}{2}$. Let $f_{\mathbf{x}}(\boldsymbol{x})$ denote the density function of a continuous random vector $\mathbf{x}$. Therefore, for any $b \in \{0, 1\}$,

$$\mathbf{Pr}\left[y = b \mid \mathbf{x} = \boldsymbol{x}\right] = \frac{\mathbf{Pr}\left[y = b\right] f_{\mathbf{x}|y}(\boldsymbol{x} \mid y = b)}{\sum_{c \in \{0,1\}} \mathbf{Pr}\left[y = c\right] f_{\mathbf{x}|y}(\boldsymbol{x} \mid y = c)} = \frac{1}{1 + \frac{f_{\mathbf{x}|y}(\boldsymbol{x} | y = 1 - b)}{f_{\mathbf{x}|y}(\boldsymbol{x} | y = b)}}.$$

Let's compute this for $b = 0$. We have

$$\frac{f_{\mathbf{x}|y}(\boldsymbol{x} \mid y = 1)}{f_{\mathbf{x}|y}(\boldsymbol{x} \mid y = 0)} = \frac{\cosh(\langle \boldsymbol{x}, \boldsymbol{\nu}\rangle / \sigma^2)}{\cosh(\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle / \sigma^2)} \exp\left(\frac{\|\boldsymbol{\mu}\|^2 - \|\boldsymbol{\nu}^2\|}{2\sigma^2}\right) = \frac{\cosh(\langle \boldsymbol{x}, \boldsymbol{\nu}\rangle / \sigma^2)}{\cosh(\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle / \sigma^2)},$$

where in the last equation we used the assumption that $\|\boldsymbol{\mu}\| = \|\boldsymbol{\nu}\|$. The decision regions are then identified by: $\mathbf{Pr}\left[y = 0 \mid \mathbf{x}\right] \geq {}^1/_2$ for label 0 and $\mathbf{Pr}\left[y = 0 \mid \mathbf{x}\right] < {}^1/_2$ for label 1.

Thus, for label 0, we need $\frac{f_{\mathbf{x}|y}(\boldsymbol{x}|y=1)}{f_{\mathbf{x}|y}(\boldsymbol{x}|y=0)} < 1$, which implies that $\frac{\cosh(\langle \boldsymbol{x}, \boldsymbol{\nu}\rangle/\sigma^2)}{\cosh(\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle/\sigma^2)} \leq 1$. Now we note that $\cosh(x) \leq \cosh(y) \implies |x| \leq |y|$ for all $x, y \in \mathbb{R}$, hence, we have $|\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle| \geq |\langle \boldsymbol{x}, \boldsymbol{\nu}\rangle|$. Similarly, we have the complementary condition for label 1. $\square$

Next, we design a two-layer and a three-layer network and show that for a particular choice of parameters $\theta = (\mathbf{W}^{(l)}, \mathbf{b}^{(l)})$ for $l \in \{1, 2\}$ for the two-layer case and $l \in \{1, 2, 3\}$ for the three-layer case, the networks realize the optimal classifier described in Lemma 3.3.1.

Proposition 3.3.2. *Consider two-layer and three-layer networks of the form described in Architecture 1, without biases ($\mathbf{b}^{(l)} = \mathbf{0}$ for all l), for parameters $\mathbf{W}^{(l)}$ and some $R \in \mathbb{R}^+$: For the two-layer network,*

$$\mathbf{W}^{(1)} = R\begin{pmatrix}\hat{\boldsymbol{\mu}} & -\hat{\boldsymbol{\mu}} & \hat{\boldsymbol{\nu}} & -\hat{\boldsymbol{\nu}}\end{pmatrix}, \qquad \mathbf{W}^{(2)} = \begin{pmatrix}-1 & -1 & 1 & 1\end{pmatrix}^\top.$$

For the three-layer network,

$$\mathbf{W}^{(1)} = R\begin{pmatrix}\hat{\boldsymbol{\mu}} & -\hat{\boldsymbol{\mu}} & \hat{\boldsymbol{\nu}} & -\hat{\boldsymbol{\nu}}\end{pmatrix}, \quad \mathbf{W}^{(2)} = \begin{pmatrix}-1 & -1 & 1 & 1\\ 1 & 1 & -1 & -1\end{pmatrix}^\top, \quad \mathbf{W}^{(3)} = \begin{pmatrix}1\\ -1\end{pmatrix}.$$

Then for any $\sigma > 0$, the defined networks realize the Bayes optimal classifier for the data model XOR-GMM$(n, \boldsymbol{\mu}, \boldsymbol{\nu}, \sigma)$.

Proof. Note that the output of the two-layer network is $\varphi([\mathbf{X}\mathbf{W}^{(1)}]_+\mathbf{W}^{(2)})$, which is interpreted as the probability with which the network believes that the input is in the class with label 1. The final prediction for the class label is assigned to be $\mathbf{1}(\hat{y}_i \geq 0.5)$. For each $i \in [n]$, we have that the output of the network on data point i is

$$\hat{y}_i = \varphi((R(|\langle \mathbf{X}_i, \hat{\boldsymbol{\nu}}\rangle| - |\langle \mathbf{X}_i, \hat{\boldsymbol{\mu}}\rangle|)),$$

where we used the fact that $[t]_+ + [-t]_+ = |t|$ for all $t \in \mathbb{R}$. Similarly, for the three-layer network, the output is $\varphi([[\mathbf{X}\mathbf{W}^{(1)}]_+\mathbf{W}^{(2)}]_+\mathbf{W}^{(3)})$. Direct calculation gives

$$\begin{aligned}\hat{y}_i &= \varphi\left(R([|\langle \mathbf{X}_i, \hat{\boldsymbol{\nu}}\rangle| - |\langle \mathbf{X}_i, \hat{\boldsymbol{\mu}}\rangle|]_+ - [|\langle \mathbf{X}_i, \hat{\boldsymbol{\mu}}\rangle| - |\langle \mathbf{X}_i, \hat{\boldsymbol{\nu}}\rangle|]_+)\right)\\ &= \varphi\left(R(|\langle \mathbf{X}_i, \hat{\boldsymbol{\nu}}\rangle| - |\langle \mathbf{X}_i, \hat{\boldsymbol{\mu}}\rangle|)\right),\end{aligned}$$

where in the last equation we used the fact that $[t]_+ - [-t]_+ = t$ for all $t \in \mathbb{R}$.

The final prediction is then obtained by considering the maximum posterior probability among the class labels 0 and 1, and thus,

$$\text{pred}(\mathbf{X}_i) = \mathbf{1}(R\,|\langle \mathbf{X}_i, \hat{\boldsymbol{\mu}}\rangle| < R\,|\langle \mathbf{X}_i, \hat{\boldsymbol{\nu}}\rangle|) = \mathbf{1}(|\langle \mathbf{X}_i, \boldsymbol{\mu}\rangle| < |\langle \mathbf{X}_i, \boldsymbol{\nu}\rangle|),$$

which matches the Bayes classifier in Lemma 3.3.1. $\square$

We are now ready to prove part one of Theorem 3. Recall from Lemma 3.3.1 that for successful classification, we require for every $i \in [n]$,

$$|\langle \mathbf{X}_i, \boldsymbol{\mu} \rangle| \geq |\langle \mathbf{X}_i, \boldsymbol{\nu} \rangle| \text{ for } i \in C_0, \qquad |\langle \mathbf{X}_i, \boldsymbol{\mu} \rangle| < |\langle \mathbf{X}_i, \boldsymbol{\nu} \rangle| \text{ for } i \in C_1.$$

Let's try to upper bound the probability of the above event, i.e., the probability that the data is classifiable. We consider only class C_0, since the analysis for C_1 is symmetric and similar. For $i \in C_0$, we can write $\mathbf{X}_i = \boldsymbol{\mu} + \sigma \mathbf{g}_i$, where $\mathbf{g}_i \sim \mathcal{N}(\mathbf{0}, I)$. Then we have for any fixed $i \in C_0$ that

$$\begin{aligned}
\mathbf{Pr}\left[|\langle \mathbf{X}_i, \boldsymbol{\mu} \rangle| \geq |\langle \mathbf{X}_i, \boldsymbol{\nu} \rangle|\right] &= \mathbf{Pr}\left[|\beta' + \sigma \langle \mathbf{g}_i, \hat{\boldsymbol{\mu}} \rangle| \geq |\sigma \langle \mathbf{g}_i, \hat{\boldsymbol{\nu}} \rangle|\right] \\
&\leq \mathbf{Pr}\left[\beta' + \sigma\, |\langle \mathbf{g}_i, \hat{\boldsymbol{\mu}} \rangle| \geq \sigma\, |\langle \mathbf{g}_i, \hat{\boldsymbol{\nu}} \rangle|\right] \quad \text{(by triangle inequality)} \\
&\leq \mathbf{Pr}\left[|\langle \mathbf{g}_i, \hat{\boldsymbol{\nu}} \rangle| - |\langle \mathbf{g}_i, \hat{\boldsymbol{\mu}} \rangle| \leq K/\sqrt{2}\right] \quad \text{(using } \beta \leq K\sigma\text{)}.
\end{aligned}$$

We now define random variables $Z_1 = \langle \mathbf{g}_i, \hat{\boldsymbol{\nu}} \rangle$ and $Z_2 = \langle \mathbf{g}_i, \hat{\boldsymbol{\mu}} \rangle$ and note that $Z_1, Z_2 \sim \mathcal{N}(0,1)$ and $\mathbb{E}[Z_1 Z_2] = 0$. Let $K' = K/\sqrt{2}$. We now have

$$\begin{aligned}
\mathbf{Pr}\left[|Z_1| - |Z_2| \leq K'\right] &= 4\mathbf{Pr}\left[Z_1 - Z_2 \leq K',\ Z_1, Z_2 \geq 0\right] \\
&= 4 \int_0^\infty \mathbf{Pr}\left[0 \leq Z_1 \leq z + K'\right] \phi(z) dz \\
&= 4 \int_0^\infty \left(\Phi(z + K') - \frac{1}{2}\right) \phi(z) dz = 4 \int_0^\infty \Phi(z + K')\, \phi(z) dz - 1 \\
&= 2\Phi(K/2) + 2\Phi(K/2)\Phi_{\mathrm{c}}(K/2) - 1 = 1 - 2\Phi_{\mathrm{c}}(K/2)^2.
\end{aligned}$$

To evaluate the integral above, we used [Owe80, Table 1:10,010.6 and Table 2:2.3]. Thus, the probability that a point $i \in C_0$ is misclassified is lower bounded as follows

$$\mathbf{Pr}\left[\mathbf{X}_i \text{ is misclassified}\right] \geq 2\Phi_{\mathrm{c}}\left(K/2\right)^2 = \tau_K.$$

Note that this is a decreasing function of K, implying that the probability of misclassification decreases as we increase the distance between the means, and is maximum for $K = 0$.

Define $M(n)$ for a fixed K to be the fraction of misclassified nodes in C_0. Define x_i to be the indicator random variable $\mathbf{1}(\mathbf{X}_i \text{ is misclassified})$. Then x_i are Bernoulli random variables with mean at least τ_K, and $\mathbb{E}M(n) = \frac{2}{n} \sum_{i \in C_0} \mathbb{E}x_i \geq \tau_K$. Using Hoeffding's inequality [Ver18, Theorem 2.2.6], we have that for any $t > 0$,

$$\mathbf{Pr}\left[M(n) \geq \tau_K - t\right] \geq \mathbf{Pr}\left[M(n) \geq \mathbb{E}M(n) - t\right] \geq 1 - \exp(-nt^2).$$

Choosing $t = n^{-\epsilon/2}$ for any $\epsilon \in (0,1)$ yields $\mathbf{Pr}\left[M(n) \geq \tau_K - n^{-\epsilon/2}\right] \geq 1 - \exp(-n^{1-\epsilon})$, which completes the proof.

We now turn to the proof of part two of Theorem 3, and show that there exists a two-layer MLP that obtains an arbitrarily small loss, and hence, successfully classifies a sample drawn from the XOR-GMM model with overwhelming probability. Consider the two-layer and three-layer MLPs described in Proposition 3.3.2, for which we have $\hat{y}_i = \varphi\left(R(|\langle \mathbf{X}_i, \hat{\boldsymbol{\nu}}\rangle| - |\langle \mathbf{X}_i, \hat{\boldsymbol{\mu}}\rangle|)\right)$.

Note that $\langle \mathbf{X}_i - \mathbb{E}\mathbf{X}_i, \hat{\boldsymbol{\mu}}\rangle$ and $\langle \mathbf{X}_i - \mathbb{E}\mathbf{X}_i, \hat{\boldsymbol{\nu}}\rangle$ are mean 0 Gaussian random variables with variance σ^2. So for any fixed $i \in [n]$ and $\mathbf{m}_c \in \{\boldsymbol{\mu}, \boldsymbol{\nu}\}$, we use [Ver18, Proposition 2.1.2] to obtain

$$\mathbf{Pr}\left[|\langle \mathbf{X}_i - \mathbb{E}\mathbf{X}_i, \hat{\mathbf{m}}_c\rangle| > t\right] \leq \frac{\sigma}{t\sqrt{2\pi}} \exp\left(-\frac{t^2}{2\sigma^2}\right).$$

Then by a union bound over all $i \in [n]$ and $\mathbf{m}_c \in \{\boldsymbol{\mu}, \boldsymbol{\nu}\}$, we have that

$$\mathbf{Pr}\left[|\langle \mathbf{X}_i - \mathbb{E}\mathbf{X}_i, \hat{\mathbf{m}}_c\rangle| \leq t \ \forall i \in [n],\ \mathbf{m}_c \in \{\boldsymbol{\mu}, \boldsymbol{\nu}\}\right] \geq 1 - \frac{n\sigma}{t}\sqrt{\frac{2}{\pi}} \exp\left(-\frac{t^2}{2\sigma^2}\right).$$

Let $\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\nu}}$ denote the normalized vectors $\boldsymbol{\mu}, \boldsymbol{\nu}$. We now set $t = \sigma\sqrt{2(c+1)\log n}$ for any large constant $c > 0$. We now have with probability at least $1 - \frac{n^{-c}}{\sqrt{\pi(c+1)\log n}}$ that

$$\langle \mathbf{X}_i, \hat{\boldsymbol{\mu}}\rangle = \langle \mathbb{E}\mathbf{X}_i, \hat{\boldsymbol{\mu}}\rangle \pm O(\sigma\sqrt{c\log n}), \quad \langle \mathbf{X}_i, \hat{\boldsymbol{\nu}}\rangle = \langle \mathbb{E}\mathbf{X}_i, \hat{\boldsymbol{\nu}}\rangle \pm O(\sigma\sqrt{c\log n}) \ \forall i \in [n].$$

Thus, we can write

$$\langle \mathbf{X}_i, \hat{\boldsymbol{\mu}}\rangle = \beta'\left(1 \pm O\left(\sqrt{\frac{c}{\log n}}\right)\right), \qquad \langle \mathbf{X}_i, \hat{\boldsymbol{\nu}}\rangle = \beta' \cdot O\left(\sqrt{\frac{c}{\log n}}\right) \qquad \forall i \in C_0, \tag{3.25}$$

$$\langle \mathbf{X}_i, \hat{\boldsymbol{\mu}}\rangle = \beta' \cdot O\left(\sqrt{\frac{c}{\log n}}\right), \qquad \langle \mathbf{X}_i, \hat{\boldsymbol{\nu}}\rangle = \beta'\left(1 \pm O\left(\sqrt{\frac{c}{\log n}}\right)\right) \qquad \forall i \in C_1. \tag{3.26}$$

Using (3.25) and (3.26), we obtain that the output of the networks in Proposition 3.3.2 is $\varphi((2y_i - 1)R\beta'(1 \pm O(\sqrt{\frac{c}{\log n}}))$, implying perfect classification of all data points.

3.3.5 Proof of Theorem 4

Networks with One Graph Convolution

Let us now prove part one of Theorem 4. First, let us understand the output of the (Bayes) optimal classifier for the XOR-GMM data model.

Lemma 3.3.3. *Let* $h(\boldsymbol{x}) = |\langle \boldsymbol{x}, \hat{\boldsymbol{\nu}}\rangle| - |\langle \boldsymbol{x}, \hat{\boldsymbol{\mu}}\rangle|$ *for all* $\boldsymbol{x} \in \mathbb{R}^d$ *and recall the value of* ζ *from* (3.24)*. Consider* $\psi(t) = t\,\mathrm{erf}(t) - \frac{1}{\sqrt{\pi}}\left(1 - e^{-t^2}\right)$*. Then we have the following.*

1. *The expectation* $\mathbb{E}h(\mathbf{X}_i) = \sqrt{2}(2y_i - 1)\sigma\psi\left(\frac{\beta}{2\sigma}\right)$*.*
2. *For any* $\beta, \sigma > 0$ *such that* $\beta = \Omega_n(\sigma)$*, we have that* $\psi(\frac{\beta}{\sigma}) = \Omega(\frac{\beta}{\sigma})$*.*
3. *For any* $\beta, \sigma > 0$ *such that* $\beta = o_n(\sigma)$*, we have that* $\psi(\frac{\beta}{\sigma}) = \Omega(\frac{\beta^2}{\sigma^2})$*.*

Proof. For part one, observe that $\langle \mathbf{X}_i, \hat{\boldsymbol{\mu}}\rangle$ and $\langle \mathbf{X}_i, \hat{\boldsymbol{\nu}}\rangle$ are Gaussian random variables with variance σ^2 and means ${}^{\beta}/_{\sqrt{2}}, 0$ if $y_i = 0$ and $0, {}^{\beta}/_{\sqrt{2}}$ if $y_i = 1$, respectively. Thus, $|\langle \mathbf{X}_i, \hat{\boldsymbol{\mu}}\rangle|$ and $|\langle \mathbf{X}_i, \hat{\boldsymbol{\nu}}\rangle|$ are folded-Gaussian random variables and we have $\mathbb{E}h(\mathbf{X}_i) = -\sqrt{2}\sigma\psi\left(\frac{\beta}{\sqrt{2}\sigma}\right)$ if $i \in C_0$ and $\mathbb{E}h(\mathbf{X}_i) = \sqrt{2}\sigma\psi\left(\frac{\beta}{\sqrt{2}\sigma}\right)$ otherwise. We now write

$$\psi(t) = t\left(\mathrm{erf}(t) - \frac{1}{t\sqrt{\pi}}(1 - e^{-t^2})\right) = tH(t),$$

where $H(t) = \mathrm{erf}(t) - \frac{1}{t\sqrt{\pi}}(1 - e^{-t^2})$.

For part two, note that $H(t)$ is an increasing function in the range $[-1, 1]$ and $H(t) > 0$ for $t > 0$. Hence, for $t \geq C$ for some positive constant C, $H(t) \geq H(C) = C'$, therefore, $\psi(t) = tH(t) \geq C't$.

For part three, $t = o_n(1)$. We use the series expansion of $h(t)$ about $t = 0$ to obtain that

$$h(t) = \frac{t}{\sqrt{\pi}} - \frac{t^3}{6\sqrt{\pi}} + O(t^5) \geq \frac{t}{\sqrt{\pi}} - \frac{t^3}{6\sqrt{\pi}} = \Omega(t).$$

Hence, $\psi(t) = th(t) = \Omega(t^2)$. Putting $t = \beta/\sigma$ completes the proof. □

Let us now begin the main proof by computing the output of the network when one graph convolution is applied at any layer other than the first.

Lemma 3.3.4. *Let* $h(\boldsymbol{x}) = |\langle \boldsymbol{x}, \hat{\boldsymbol{\nu}}\rangle| - |\langle \boldsymbol{x}, \hat{\boldsymbol{\mu}}\rangle|$ *for any* $\boldsymbol{x} \in \mathbb{R}^d$*. Consider the two-layer and three-layer networks in Proposition 3.3.2 where the weight parameter of the last layer,* $\mathbf{W}^{(L)}$*, is scaled by a factor of* $\xi = \operatorname{sgn}(p-q)$*. If a graph convolution is added to these networks in either the second or the third layer then for a sample* $(\mathbf{A}, \mathbf{X}) \sim$ *XOR-CSBM*$(n, p, q, \boldsymbol{\mu}, \boldsymbol{\nu}, \sigma)$*, the output of the networks for a point* $i \in [n]$ *is*

$$\hat{y}_i = \varphi(f_i^{(L)}(\mathbf{X})) = \varphi\left(\frac{R\operatorname{sgn}(p-q)}{\mathbf{deg}(i)}\sum_{j\in[n]} a_{ij}h(\mathbf{X}_j)\right).$$

Proof. The networks with scaled parameters are given as follows.
For the two-layer network:

$$\mathbf{W}^{(1)} = R\begin{pmatrix}\hat{\boldsymbol{\mu}} & -\hat{\boldsymbol{\mu}} & \hat{\boldsymbol{\nu}} & -\hat{\boldsymbol{\nu}}\end{pmatrix}, \qquad \mathbf{W}^{(2)} = \xi\begin{pmatrix}-1 & -1 & 1 & 1\end{pmatrix}^\top.$$

For the three-layer network:

$$\mathbf{W}^{(1)} = R\xi\begin{pmatrix}\hat{\boldsymbol{\mu}} & -\hat{\boldsymbol{\mu}} & \hat{\boldsymbol{\nu}} & -\hat{\boldsymbol{\nu}}\end{pmatrix}, \quad \mathbf{W}^{(2)} = \begin{pmatrix}-1 & -1 & 1 & 1\\ 1 & 1 & -1 & -1\end{pmatrix}^\top, \quad \mathbf{W}^{(3)} = \xi\begin{pmatrix}1\\ -1\end{pmatrix}.$$

When a graph convolution is applied at the second layer of this two-layer MLP, the output of the last layer for data $(\mathbf{A}, \mathbf{X})$ is $f_i^{(2)}(\mathbf{X}) = \mathbf{D}^{-1}\mathbf{A}[\mathbf{X}\mathbf{W}^{(1)}]_+\mathbf{W}^{(2)}$. Then we have

$$f_i^{(2)}(\mathbf{X}) = \frac{R\xi}{\mathbf{deg}(i)}\sum_{j\in[n]} a_{ij}(|\langle \mathbf{X}_j, \hat{\boldsymbol{\nu}}\rangle| - |\langle \mathbf{X}_j, \hat{\boldsymbol{\mu}}\rangle|) = \frac{R\xi}{\mathbf{deg}(i)}\sum_{j\in[n]} a_{ij}h(\mathbf{X}_j).$$

Similarly, when the graph convolution is applied at the second layer of the three-layer MLP, the output is $f_i^{(3)}(\mathbf{X}) = [\mathbf{D}^{-1}\mathbf{A}[\mathbf{X}\mathbf{W}^{(1)}]_+\mathbf{W}^{(2)}]_+\mathbf{W}^{(3)}$, and we have

$$f_i^{(3)}(\mathbf{X}) = \frac{R\xi}{\mathbf{deg}(i)}\left(\left[\sum_{j\in[n]} a_{ij}h(\mathbf{X}_j)\right]_+ - \left[-\sum_{j\in[n]} a_{ij}h(\mathbf{X}_j)\right]_+\right) = \frac{R\xi}{\mathbf{deg}(i)}\sum_{j\in[n]} a_{ij}h(\mathbf{X}_j).$$

Finally, when the graph convolution is applied at the third layer of the three-layer MLP, the output is $f_i^{(3)}(\mathbf{X}) = \mathbf{D}^{-1}\mathbf{A}[[\mathbf{X}\mathbf{W}^{(1)}]_+\mathbf{W}^{(2)}]_+\mathbf{W}^{(3)}$, and we have

$$\begin{aligned} f_i^{(3)}(\mathbf{X}) &= \frac{R\xi}{\mathbf{deg}(i)}\sum_{j\in[n]} a_{ij}\left([|\langle \mathbf{X}_j, \hat{\boldsymbol{\nu}}\rangle| - |\langle \mathbf{X}_j, \hat{\boldsymbol{\mu}}\rangle|]_+ - [|\langle \mathbf{X}_j, \hat{\boldsymbol{\mu}}\rangle| - |\langle \mathbf{X}_j, \hat{\boldsymbol{\nu}}\rangle|]_+\right)\\ &= \frac{R}{\mathbf{deg}(i)}\sum_{j\in[n]} a_{ij}(|\langle \mathbf{X}_j, \hat{\boldsymbol{\nu}}\rangle| - |\langle \mathbf{X}_j, \hat{\boldsymbol{\mu}}\rangle|) = \frac{R\xi}{\mathbf{deg}(i)}\sum_{j\in[n]} a_{ij}h(\mathbf{X}_j). \end{aligned}$$

Therefore, in all cases with a single convolution, the output of the last layer is $f_i^{(L)}(\mathbf{X}) = \frac{R\,\mathrm{sgn}(p-q)}{\mathbf{deg}(i)}\sum_{j\in[n]} a_{ij} h(\mathbf{X}_j)$, where $L \in \{2,3\}$ is the number of layers. $\square$

Now to complete the proof of Theorem 4 part one, we analyze the output conditioned on the adjacency matrix $\mathbf{A}$. Note that $\frac{1}{R} f_i^{(L)}(\mathbf{X})$ in Lemma 3.3.4 is Lipschitz with constant $\sqrt{\frac{2}{\mathbf{deg}(i)}}$, and $h(\mathbf{X}_j)$ are mutually independent for $j \in [n]$. Therefore, by Gaussian concentration [Ver18, Theorem 5.2.2] we have that for a fixed $i \in [n]$,

$$\mathbf{Pr}\left[\frac{1}{R}|f_i^{(L)}(\mathbf{X}) - \mathbb{E}[f_i^{(L)}(\mathbf{X})]| > \delta \mid \mathbf{A}\right] \le 2\exp\left(-\frac{\delta^2 \mathbf{deg}(i)}{4\sigma^2}\right).$$

We refer to the event from Proposition 2.4.6 as B and define $Q(t)$ to be the event that $|f_i^{(L)}(\mathbf{X}) - \mathbb{E}[f_i^{(L)}(\mathbf{X})]| \le t$ for all $i \in [n]$. Then we can write

$$\begin{aligned}\mathbf{Pr}\left[Q(t)^{\mathsf{c}}\right] = \mathbf{Pr}\left[Q(t)^{\mathsf{c}} \cap B\right] + \mathbf{Pr}\left[Q(t)^{\mathsf{c}} \cap B^{\mathsf{c}}\right] &\le 2n\exp\left(-\frac{t^2 n(p+q)}{8\sigma^2}\right) + \mathbf{Pr}\left[B^{\mathsf{c}}\right] \\ &\le 2n\exp\left(-\frac{t^2 n(p+q)}{8\sigma^2}\right) + 2n^{-c}.\end{aligned}$$

Let $\xi = \mathrm{sgn}(p-q)$ and note that $\frac{\xi(p-q)}{p+q} = \frac{|p-q|}{p+q} = \gamma(p,q)$. We now choose $t = 2\sigma\sqrt{\frac{2(c+1)\log n}{n(p+q)}}$ to obtain that with probability at least $1-4n^{-c}$, the following holds for all $i \in [n]$:

$$\begin{aligned}\frac{1}{\sigma} f_i^{(L)}(\mathbf{X}) &= \frac{1}{\sigma}\mathbb{E}[f_i^{(L)}(\mathbf{X})] \pm O\left(R\sqrt{\frac{c\log n}{n(p+q)}}\right) \\ &= \frac{R\xi}{\sigma\,\mathbf{deg}(i)}\sum_{j\in[n]} a_{ij}\mathbb{E}h(\mathbf{X}_j) \pm O\left(R\sqrt{\frac{c\log n}{n(p+q)}}\right) \\ &= \frac{\sqrt{2}R\xi\zeta}{\sigma\,\mathbf{deg}(i)}\left(\sum_{j\in C_1} a_{ij} - \sum_{j\in C_0} a_{ij}\right) \pm O\left(R\sqrt{\frac{c\log n}{n(p+q)}}\right) && \text{(Lemma 3.3.3)} \\ &= \sqrt{2}(2y_i - 1)R\gamma(p,q)\zeta(1 \pm o_n(1)) \pm O\left(R\sqrt{\frac{c\log n}{n(p+q)}}\right) && \text{(\textit{Proposition} 2.4.6)} \\ &= \sqrt{2}(2y_i - 1)R\gamma(p,q)\zeta(1 \pm o_n(1)),\end{aligned}$$

where in the last equation we used the assumption that $\gamma(p,q)\zeta = \omega\left(\sqrt{\frac{\log n}{n(p+q)}}\right)$. Overall, we obtain that with probability at least $1-4n^{-c}$,

$$f_i^{(L)}(\mathbf{X}) = (2y_i - 1)C\sigma R\gamma(p,q)\zeta(1 \pm o_n(1)), \text{ for all } i \in [n].$$

Therefore, with probability at least $1 - O(n^{-c})$, the output of the networks have correct signs for each class, implying perfect classification of all the nodes.

Networks with Two Graph Convolutions

Let us now turn to the proof of part two of Theorem 4. We begin by computing the output of the networks constructed in Proposition 3.3.2 when two graph convolutions are placed among any layer in the networks other than the first.

Lemma 3.3.5. *Let* $h(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R} = |\langle \boldsymbol{x}, \hat{\boldsymbol{\nu}} \rangle| - |\langle \boldsymbol{x}, \hat{\boldsymbol{\mu}} \rangle|$*. Consider the networks constructed in Proposition 3.3.2 equipped with two graph convolutions in the following combinations:*

1. *Both convolutions in the second layer of the two-layer network.*
2. *Both convolutions in the second layer of the three-layer network.*
3. *One convolution each in the second and the third layer of the three-layer network.*
4. *Both convolutions in the third layer of the three-layer network.*

Then for a sample $(\mathbf{A}, \mathbf{X}) \sim \textit{XOR-CSBM}(n, p, q, \boldsymbol{\mu}, \boldsymbol{\nu}, \sigma)$*, the output of the networks in all the above described combinations for a point* $i \in [n]$ *is*

$$\hat{y}_i = \varphi(f_i^{(L)}(\mathbf{X})) = \varphi\left(\frac{R}{\mathbf{deg}(i)} \sum_{j \in [n]} \tau_{ij} h(\mathbf{X}_j)\right), \text{ where } \tau_{ij} = \sum_{k \in [n]} \frac{a_{ik} a_{jk}}{\mathbf{deg}(k)}.$$

Proof. For the two-layer network, the output of the last layer when both convolutions are at the second layer is given by $f_i^{(2)}(\mathbf{X}) = (\mathbf{D}^{-1}\mathbf{A})^2 [\mathbf{X}\mathbf{W}^{(1)}]_+ \mathbf{W}^{(2)}$. Then we have

$$f_i^{(2)}(\mathbf{X}) = \frac{R}{\mathbf{deg}(i)} \sum_{j \in [n]} \sum_{k \in [n]} \frac{a_{ij} a_{jk}}{\mathbf{deg}(j)} h(\mathbf{X}_k) = \frac{R}{\mathbf{deg}(i)} \sum_{j \in [n]} \tau_{ij} h(\mathbf{X}_j).$$

Next, for the three-layer network, the output of the last layer when both convolutions are at the second layer is given by $f_i^{(3)}(\mathbf{X}) = [(\mathbf{D}^{-1}\mathbf{A})^2 [\mathbf{X}\mathbf{W}^{(1)}]_+ \mathbf{W}^{(2)}]_+ \mathbf{W}^{(3)}$, hence, we have

$$f_i^{(3)}(\mathbf{X}) = \frac{R}{\mathbf{deg}(i)} \left(\left[\sum_{j \in [n]} \frac{a_{ij}}{\mathbf{deg}(j)} \sum_{k \in [n]} a_{jk} h(\mathbf{X}_k) \right]_+ - \left[- \sum_{j \in [n]} \frac{a_{ij}}{\mathbf{deg}(j)} \sum_{k \in [n]} a_{jk} h(\mathbf{X}_k) \right]_+ \right)$$

$$= \frac{R}{\mathbf{deg}(i)} \sum_{j\in[n]} \frac{a_{ij}}{\mathbf{deg}(j)} \sum_{k\in[n]} a_{jk} h(\mathbf{X}_k) = \frac{R}{\mathbf{deg}(i)} \sum_{j\in[n]} \tau_{ij} h(\mathbf{X}_j).$$

Similarly, the output of the last layer when there are one convolution each at the second and the third layer is given by $f_i^{(3)}(\mathbf{X}) = \mathbf{D}^{-1}\mathbf{A}[\mathbf{D}^{-1}\mathbf{A}[\mathbf{X}\mathbf{W}^{(1)}]_+\mathbf{W}^{(2)}]_+\mathbf{W}^{(3)}$, hence, we have

$$\begin{aligned} f_i^{(3)}(\mathbf{X}) &= \frac{R}{\mathbf{deg}(i)} \sum_{j\in[n]} \frac{a_{ij}}{\mathbf{deg}(j)} \left(\left[\sum_{k\in[n]} a_{jk} h(\mathbf{X}_k) \right]_+ - \left[-\sum_{k\in[n]} a_{jk} h(\mathbf{X}_k) \right]_+ \right) \\ &= \frac{R}{\mathbf{deg}(i)} \sum_{j\in[n]} \frac{a_{ij}}{\mathbf{deg}(j)} \sum_{k\in[n]} a_{jk} h(\mathbf{X}_k) = \frac{R}{\mathbf{deg}(i)} \sum_{j\in[n]} \tau_{ij} h(\mathbf{X}_j). \end{aligned}$$

Finally, the output of the last layer when both convolutions are at the third layer is given by $f_i^{(3)}(\mathbf{X}) = (\mathbf{D}^{-1}\mathbf{A})^2[[\mathbf{X}\mathbf{W}^{(1)}]_+\mathbf{W}^{(2)}]_+\mathbf{W}^{(3)}$, hence, we have

$$\begin{aligned} f_i^{(3)}(\mathbf{X}) &= \frac{R}{\mathbf{deg}(i)} \sum_{j\in[n]} \frac{a_{ij}}{\mathbf{deg}(j)} \left(\sum_{k\in[n]} a_{jk} \left([h(\mathbf{X}_k)]_+ - [-h(\mathbf{X}_k)]_+ \right) \right) \\ &= \frac{R}{\mathbf{deg}(i)} \sum_{j\in[n]} \frac{a_{ij}}{\mathbf{deg}(j)} \sum_{k\in[n]} a_{jk} h(\mathbf{X}_k) = \frac{R}{\mathbf{deg}(i)} \sum_{j\in[n]} \tau_{ij} h(\mathbf{X}_j). \end{aligned}$$

Hence, the output for two graph convolutions is the same for any combination of the placement of convolutions, as long as no convolution is placed at the first layer. □

We are now ready to prove the positive result for two convolutions. The proof strategy is similar to that of part one of the theorem. Note that $\frac{1}{R} f_i^{(L)}(\mathbf{X})$ in Lemma 3.3.5 is Lipschitz with constant

$$\left\| \frac{1}{R} f_i^{(L)}(\mathbf{X}) \right\|_{\text{Lip}} \leq \sqrt{\frac{2}{\mathbf{deg}(i)^2} \sum_{j\in[n]} \tau_{ij}^2}.$$

Since $h(\mathbf{X}_j)$ are mutually independent for $j \in [n]$, by Gaussian concentration [Ver18, Theorem 5.2.2] we have that for a fixed $i \in [n]$,

$$\mathbf{Pr}\left[\frac{1}{R} |f_i^{(L)}(\mathbf{X}) - \mathbb{E}[f_i^{(L)}(\mathbf{X})]| > \delta \mid \mathbf{A} \right] \leq 2\exp\left(-\frac{\delta^2 \mathbf{deg}(i)^2}{4\sigma^2 \sum_{j\in[n]} \tau_{ij}^2} \right).$$

We refer to the event from Proposition 2.4.7 as B. Note that since the graph density assumption is stronger than $\Omega(\frac{\log^2 n}{n})$, Proposition 2.4.6 trivially holds in this regime, hence, the degrees also concentrate strongly around $\Delta = \frac{n}{2}(p+q)$. On event B, we have that

$$\begin{aligned}
\sum_{j\in[n]} \tau_{ij}^2 &= \sum_{j\in[n]} \left(\sum_{k\in[n]} \frac{a_{ik}a_{jk}}{\mathbf{deg}(k)} \right)^2 = \frac{1}{\Delta^2} \sum_{j\in[n]} \left(\sum_{k\in[n]} a_{ik}a_{jk} \right)^2 (1 \pm o_n(1)) \\
&= \frac{1}{\Delta^2} \left(\sum_{j\sim i} |N_i \cap N_j|^2 + \sum_{j\nsim i} |N_i \cap N_j|^2 \right) (1 \pm o_n(1)) \\
&= \frac{1}{\Delta^2} \left(\sum_{j\sim i} \left(\frac{n}{2}(p^2+q^2) \right)^2 + \sum_{j\nsim i} (npq)^2 \right) (1 \pm o_n(1)) \qquad \text{(using Proposition 2.4.7)} \\
&= \frac{n}{2\Delta^2} \left(\frac{n^2}{4}(p^2+q^2)^2 + n^2p^2q^2 \right) (1 \pm o_n(1)) = \frac{n^3}{8\Delta^2} \left(p^4 + q^4 + 6p^2q^2 \right) (1 \pm o_n(1)).
\end{aligned}$$

Therefore, under this event we have that

$$\left\| \frac{1}{R} f_i^{(L)}(\mathbf{X}) \right\|_{\mathrm{Lip}} \leq \sqrt{\frac{2}{\mathbf{deg}(i)^2} \sum_{j\in[n]} \tau_{ij}^2} = \sqrt{\frac{4(p^4+q^4+6p^2q^2)}{n(p+q)^4}} (1 \pm o_n(1)).$$

Note that $K = K(p,q) = \frac{4(p^4+q^4+6p^2q^2)}{(p+q)^4} \leq 4$. We now define $Q(t)$ to be the event that $|f_i^{(L)}(\mathbf{X}) - \mathbb{E}[f_i^{(L)}(\mathbf{X})]| \leq t$ for all $i \in [n]$. Then we have

$$\mathbf{Pr}\left[Q(t)^{\mathsf{c}}\right] = \mathbf{Pr}\left[Q(t)^{\mathsf{c}} \cap B\right] + \mathbf{Pr}\left[Q(t)^{\mathsf{c}} \cap B^{\mathsf{c}}\right] \leq 2n \exp\left(-\frac{nt^2}{2K\sigma^2} \right) + 2n^{-c}.$$

We now choose $t = \sigma\sqrt{\frac{2K(c+1)\log n}{n}}$ to obtain that with probability at least $1 - 4n^{-c}$, the following holds for all $i \in [n]$:

$$f_i^{(L)}(\mathbf{X}) = \mathbb{E}[f_i^{(L)}(\mathbf{X})] \pm O\left(R\sigma\sqrt{\frac{\log n}{n}} \right) = \frac{R}{\mathbf{deg}(i)} \sum_{j\in[n]} \tau_{ij} \mathbb{E}h(\mathbf{X}_j) \pm O\left(R\sigma\sqrt{\frac{\log n}{n}} \right).$$

Note that we have

$$\frac{1}{\sigma\mathbf{deg}(i)} \sum_{j\in[n]} \tau_{ij} \mathbb{E}h(\mathbf{X}_j) = \frac{\sqrt{2}\zeta}{\mathbf{deg}(i)} \left(\sum_{j\in C_1} \tau_{ij} - \sum_{j\in C_0} \tau_{ij} \right) \qquad \text{(using Lemma 3.3.3)}$$

$$
\begin{aligned}
&= \frac{\sqrt{2}\zeta}{\mathbf{deg}(i)} \left(\sum_{j\in C_1} \sum_{k\in[n]} \frac{a_{ik}a_{jk}}{\mathbf{deg}(k)} - \sum_{j\in C_0} \sum_{k\in[n]} \frac{a_{ik}a_{jk}}{\mathbf{deg}(k)} \right) \\
&= \frac{\sqrt{2}\zeta}{\mathbf{deg}(i)} \left(\sum_{k\in[n]} \frac{a_{ik}}{\mathbf{deg}(k)} \left(\sum_{j\in C_1} a_{jk} - \sum_{j\in C_0} a_{jk} \right) \right) \\
&= \frac{\sqrt{2}\zeta \operatorname{sgn}(p-q)\gamma(p,q)}{\mathbf{deg}(i)} \left(\sum_{k\in C_1} a_{ik} - \sum_{k\in C_0} a_{ik} \right) (1+o_n(1)) \\
&= \sqrt{2}(2y_i - 1)\zeta\gamma(p,q)^2(1+o_n(1)).
\end{aligned}
$$

In the last two equations above, we used Proposition 2.4.6. Overall, we obtain that

$$
\begin{aligned}
\frac{1}{\sigma} f_i^{(L)}(\mathbf{X}) &= (2y_i - 1)CR\gamma(p,q)^2\zeta(1+o_n(1)) \pm O\left(R\sqrt{\frac{\log n}{n}}\right) \\
&= (2y_i - 1)CR\gamma(p,q)^2\zeta(1+o_n(1)),
\end{aligned}
$$

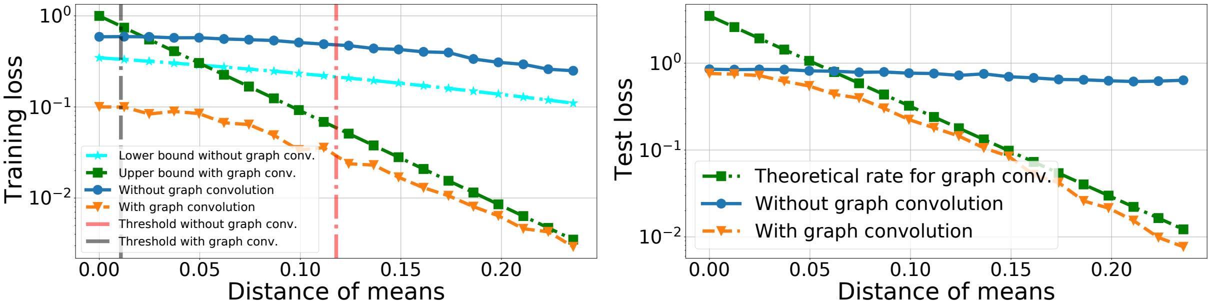
where in the last equation we used $\gamma(p,q)^2\zeta = \omega\left(\sqrt{\frac{\log n}{n}}\right)$. Now that the outputs have the correct sign, this implies perfect classification of all nodes and completes the proof.

3.4 Experiments with the Linear Model

In this section, I provide experiments to demonstrate the theoretical results in Section 3.2. To solve problem (3.1), CVX was used, a package for specifying and solving convex programs [GB13, BBK08].

3.4.1 Loss against the Distance between the Means

In our first experiment, we illustrate how the training and test losses scale as the distance between the means increases from nearly zero to $2/\sqrt{d}$. Note that according to Part 1 and Part 3 of Theorem 1, $1/\sqrt{0.5dn(p+q)}$ and $1/\sqrt{d}$ are the thresholds for the distance between the means, below which the data with and without graph convolution are not linearly separable with high probability, respectively. For this experiment, we train and test on a CSBM with $p = 0.5$, $q = 0.1$, $d = 60$, and $n = 400$ which is roughly equal to $0.85 \cdot d^{3/2}$, and each class has 200 nodes. We present results averaged over 10 trials for the training data and 10 trials for the test data. This means that for each value of the distance between the means we have 100 combinations of train and test data. The results for training loss are shown in Fig. 3.2a and the results of the test loss are shown in Fig. 3.2b. We observe that graph convolution results in smaller training and test loss when the distance of the means is larger than $\log n/\sqrt{dn} \approx 0.035$, which is the threshold such that graph convolution can linearly separate the data (Part 2 of Theorem 1).



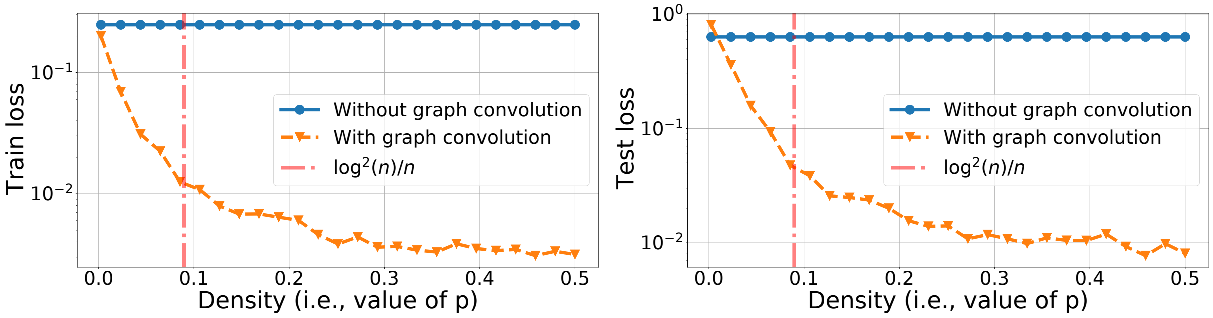
(a) Training loss vs distance of means

(b) Test loss vs distance of means

Figure 3.2: Training and test loss with/without graph convolution for increasing distance between the means. The vertical dashed red and black lines correspond to the separability thresholds from Parts 1 and 3 of Theorem 1, respectively. The green dashed line with square markers illustrates the theoretical rate from Theorem 2. The cyan dashed line with star markers corresponds to the lower bound from Part 1 of Theorem 1. We train and test on a CSBM with $p = 0.5$, $q = 0.1$, $n = 400$ and $d = 60$. The y-axis is in the log scale.

3.4.2 Loss against the Density of the Graph

In our second experiment, we illustrate how the training and test losses scale as the density of the graph increases while maintaining the same signal-to-noise ratio for the graph. By density we mean the value of the intra- and inter-class edge probabilities p and q, since they both control the average degree of each node in the graph. For this experiment, we train and test on a CSBM with $q = 0.2p$ where p varies from $1/n$ to 0.5 and $\gamma(p,q) \approx 0.6$, $d = 60$, and $n = 400$ which is roughly equal to $0.85 \cdot d^{3/2}$, and each class has 200 nodes. For this experiment, the distance between the means is $2/\sqrt{d}$. The results for training loss are shown in Fig. 3.3a and the results of the test loss are shown in Fig. 3.3b. In these figures, we observe that the performance of graph convolution improves as density increases. We also observe that for $p, q \leq \log^2 n/n$, the performance of graph convolution is as poor as that of standard logistic regression.



(a) Training loss vs density

(b) Test loss vs density

Figure 3.3: Training and test loss with/without graph convolution for increasing density. The vertical dashed red line corresponds to the lower bound of p and q from our assumption on graph sparsity. The y-axis is in the log scale.

3.4.3 Out-of-Distribution Generalization on Synthetic Data

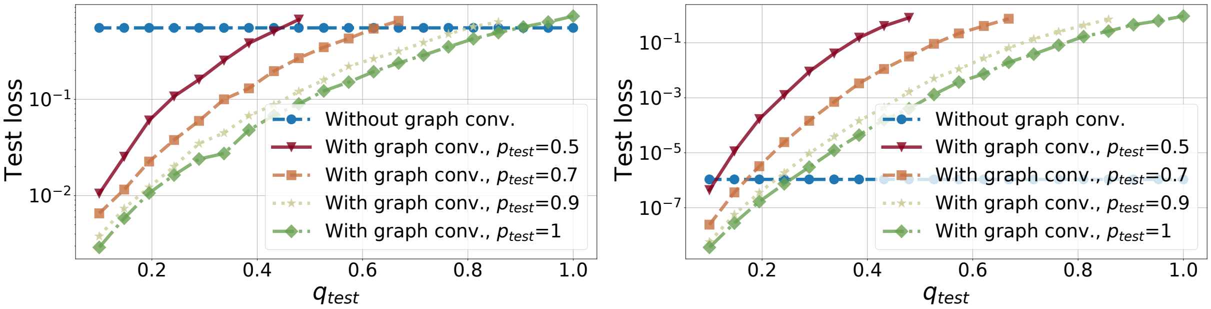
In this experiment, we test the performance of the trained classifier on out-of-distribution datasets. We perform this experiment for two different distances between the means, $16/\sqrt{d}$ and $2/\sqrt{d}$. We train on a CSBM with $p_{train} = 0.5$, $q_{train} = 0.1$, $n = 400$ and $d = 60$, and we test on CSBMs with $n = 400$, $d = 60$ and varying p_{test} and q_{test} while $p_{test} > q_{test}$. The results are shown in Fig. 3.4[1]. In this figure, we observe what was studied in Theorem 2 that is, out-of-distribution generalization to CSBMs with the same means but different p and q pairs. In particular, for a small distance between the means, i.e., $2/\sqrt{d}$, where the data are close to being not linearly separable with high probability (Part 1 Theorem 1), Fig. 3.4a shows that graph convolution results in much lower test error than not using the graph. This happens even when q_{test} is close to p_{test} in the figure, i.e., $\gamma(p_{test}, q_{test})$ from the bound in Theorem 2 is small. Furthermore, in Fig. 3.4b, we observe that for large distance between the means, i.e., $16/\sqrt{d}$, where the data are linearly separable with high probability (Part 1 Theorem 1), and q_{test} is much smaller than p_{test} (i.e., $\gamma(p_{test}, q_{test})$ is large), then graph convolution has low test error, and this error is lower than that obtained without using the graph. On the other hand, in this regime for the means, as q_{test} approaches p_{test}

---

[1] Note that the x-axis is q. Another option that is more aligned with Theorem 2 is $\gamma(p_{test}, q_{test})$, however, the log-scale collapses all lines to one and the result is less visually informative.

(i.e, as $\gamma(p_{test}, q_{test})$ decreases), the test error increases and eventually it becomes larger than without the graph.

In summary, we observe that in the difficult regime where the data are close to linearly inseparable, i.e., the means are close but larger than $1/\sqrt{d}$, then graph convolution can be very beneficial. However, if the data are linearly separable and their means are far apart, then we get good performance without the graph. Furthermore, if $\gamma(p_{test}, q_{test})$ is small then the graph convolution can result in worse training and test errors than logistic regression on the data alone. In the supplementary material, we provide similar plots for various training pairs p_{test} and q_{test}. We observe similar trends in those experiments.



(a) $\|\mu_d - \nu_d\| = 2/\sqrt{d}$

(b) $\|\mu_d - \nu_d\| = 16/\sqrt{d}$

Figure 3.4: Out-of-distribution generalization. We train on a CSBM with $p_{train} = 0.5$, $q_{train} = 0.1$, $n = 400$ and $d = 60$. We test on CSBMs with $n = 400$, $d = 60$ and varying p_{test} and q_{test} while $p_{test} > q_{test}$ and fixed means. The y-axis is in the log scale.

3.4.4 Out-of-Distribution Generalization on Real Data

In this experiment, we illustrate the generalization performance on real data for the linear classifier obtained by solving (3.1). Note that Eq. (3.1) is convex in the case of one-layer networks. In particular, we use the partially labelled real data to train two linear classifiers, with and without graph convolution. We generate new graphs by adding inter-class edges uniformly at random. Then we test the performance of the trained classifiers on the noisy graphs with the original attributes. Therefore, the only thing that changes in the new unseen data are the graphs, the attributes remain the same. Note that our goal in this experiment is not to beat current baselines, but rather to demonstrate out-of-distribution generalization for real data when we use graph convolution.

We use the real datasets Cora, PubMed and WikipediaNetwork. These datasets are publicly available and can be downloaded from [FL19]. The datasets come with multiple classes, however, for each of our experiments, we do a one-v.s.-all classification for a single class. WikipediaNetwork comes with multiple masks for the labels, in our experiments, we use the first mask. Moreover, this is a semi-supervised problem, meaning that only a fraction of the training nodes have labels. Details about the datasets are given in Table 3.1.

Table 3.1: Information about the datasets, β_0 and β_1 are defined in Section 2.2. Note that for each dataset we only consider classes A and B and we perform linear classification in a one-v.s.-all fashion. Here, A and B refer to the original classes of the dataset. Results for other classes are given in the supplementary material.

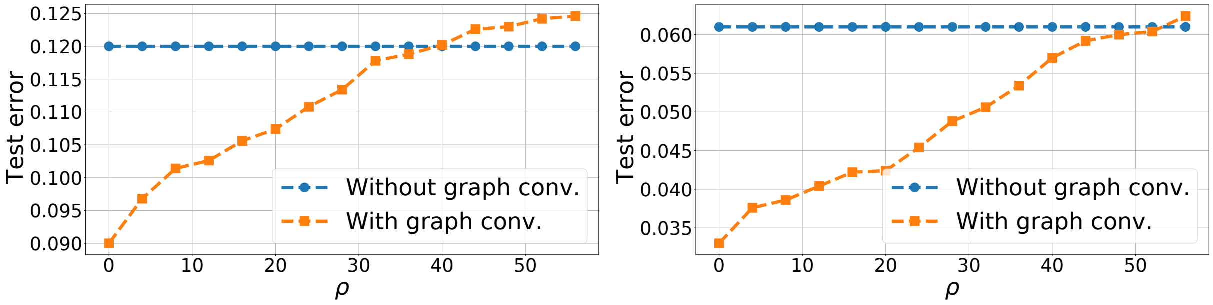
Info./Dataset	Cora	PubMed	Wiki.Net.
# nodes	2708	19717	2277
# attributes	1433	500	2325
β_0, class A	5.0e−2	2.5e−3	4.7e−1
β_1, class A	5.6e−2	4.8e−3	4.9e−1
β_0, class B	4.8e−2	3.3e−3	4.7e−1
β_1, class B	9.2e−2	2.5e−3	4.7e−1
$\|\boldsymbol{\mu} - \boldsymbol{\nu}\|$, class A	7.0e−1	1.0e−1	3.6e−1
$\|\boldsymbol{\mu} - \boldsymbol{\nu}\|$, class B	9.4e−1	7.2e−2	3.0e−1

The results of this experiment are presented in Fig. 3.5. We present results for classes A and B for each dataset. This set of experiments is enough to demonstrate good and bad performance when using graph convolution. The results for the rest of the classes are presented in the supplementary material. The performance for other classes is similar. Note in the plots that in this figure the y-axis (Test error) measures the number of misclassified nodes[2] over the number of nodes in the graph. In all sub-figures in Fig. 3.5 except for Fig. 3.5c we observe that graph convolution has lower test error than without the graph convolution. However, as we add inter-class edges (noise increases), then graph convolution can be disadvantageous. Also, there can be cases like in Fig. 3.5c where graph convolution is disadvantageous for any level of noise. Interestingly, in the experiment in Fig. 3.5c the

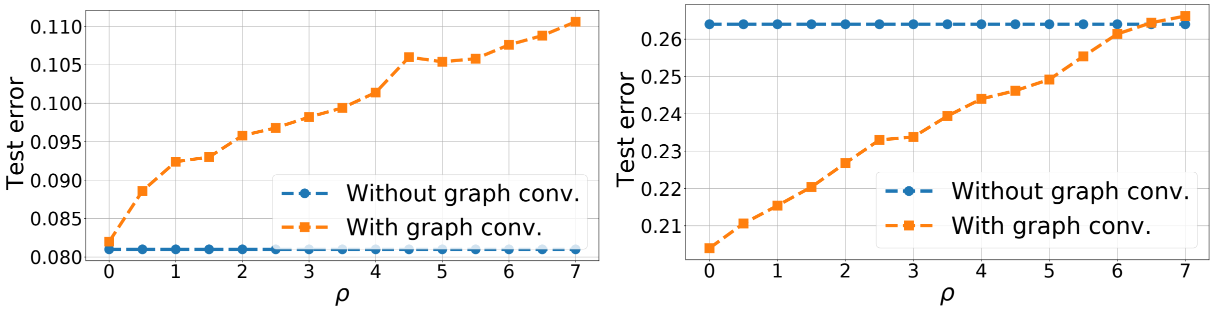
---

[2] We do not plot the loss on the y-axis because the test loss does not differ much between using and not using graph convolution. However, the number of misclassified nodes differs significantly as shown in Fig. 3.5. As noted after Theorem 2, our argument for the bound on the loss immediately yields a bound on the number of misclassified nodes.

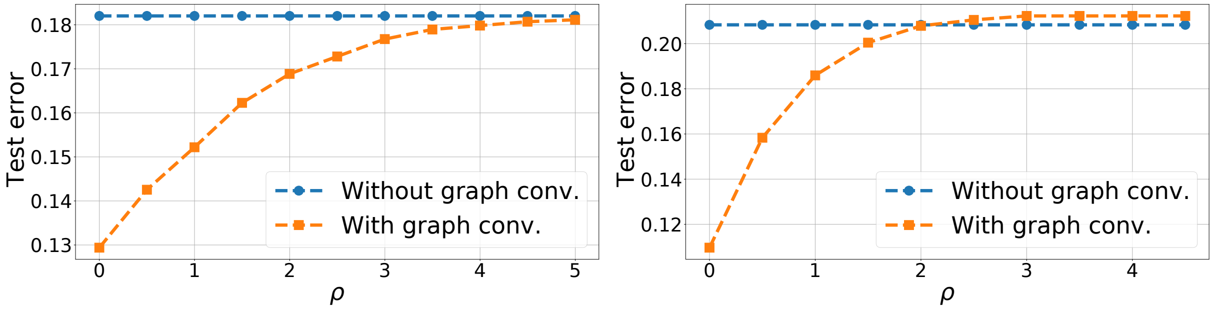
test errors with and without graph convolution are low (roughly ~ 0.080). This seems to imply that the dataset is close to being linearly separable for the given labels. However, the dataset seems to be nearly non-separable after the graph convolution, since adding noise to the graph results in a larger test error.



(a) Cora, class A (b) Cora, class B



(c) PubMed, class A (d) PubMed, class B



(e) WikipediaNetwork, class A (f) WikipediaNetwork, class B

Figure 3.5: Test loss as the number of nodes increases. The test error measures the number of misclassified nodes over the number of nodes in the graph. Moreover, ρ denotes the ratio of added inter-class edges over the number of inter-class edges of the original graph. The y-axis is in the log scale.

3.5 Experiments with the Multi-layer Models

In this section, I provide empirical evidence that supports our results in Section 3.3. We begin by analyzing the synthetic data models XOR-GMM and XOR-CSBM that are crucial to our theoretical results, followed by a similar analysis on multiple real-world datasets tailored for node classification tasks. I show a comparison of the test accuracy obtained by various learning methods in different regimes, along with a display of how the performance changes with the properties of the underlying graph, i.e., with the intra-class and inter-class edge probabilities p and q.

We observe that the performance of the networks does not change significantly with the choice of the placement of graph convolutions. In particular, placing all convolutions in the last layer achieves a similar performance as any other placement for the same number of convolutions. This observation aligns with the results in [GBG19].

3.5.1 Synthetic Data

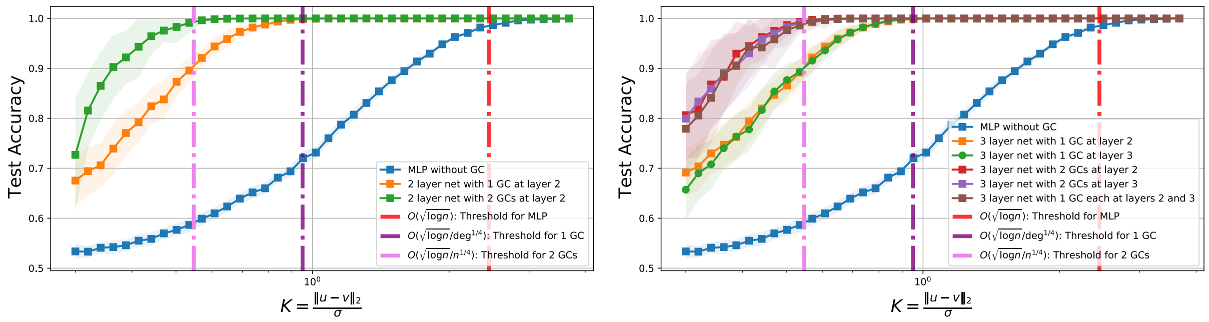
In this section, we empirically show the landscape of the accuracy achieved for various multi-layer networks with up to three layers and up to two graph convolutions. The reader is referred to Section 3.6.3 for experiments with networks having graph convolutions in the first layer, which provide insights into why convolutions may hurt in the first layer.

In Fig. 3.6, we see that as claimed in Theorem 4, a single graph convolution reduces the classification threshold by a factor of ${}^{1}\!/\!\sqrt[4]{\mathbb{E}\,\mathrm{deg}}$ and two graph convolutions reduce the threshold by a factor of ${}^{1}\!/\!\sqrt[4]{n}$, where $\mathbb{E}\,\mathbf{deg} = \frac{n}{2}(p+q)$.

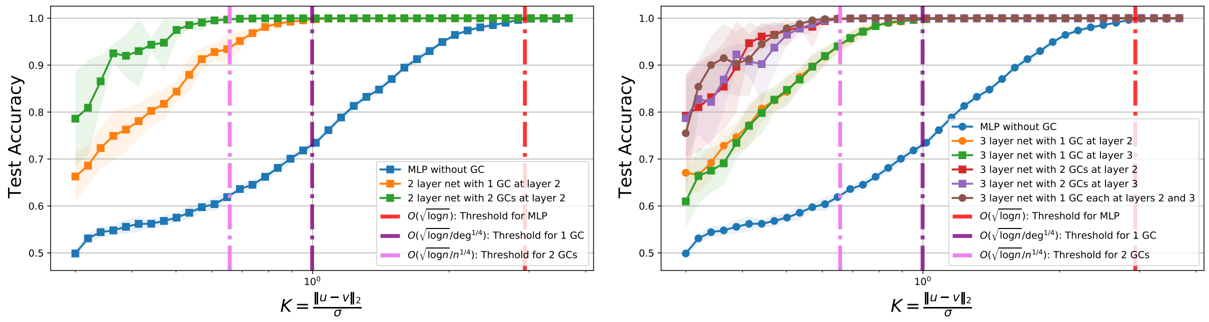
We observe that the placement of graph convolutions does not matter as long as it is not in the first layer. Figs. 3.6a and 3.6b show that the performance is mutually similar for all networks that have one graph convolution placed in the second or the third layer, and for all networks that have two graph convolutions placed in any combination among the second and the third layers. In Figs. 3.6e and 3.6f, we observe that two graph convolutions do not obtain a significant advantage over one graph convolution in the setting where p and q are large, i.e., when the graph is dense.

3.5.2 Real-world Data

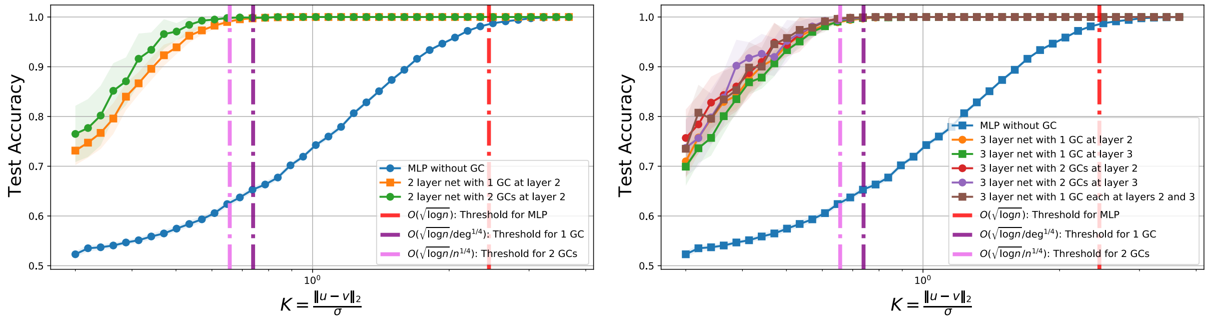
For real-world data, we test our results on three graph benchmarks: *CORA*, *CiteSeer*, and *Pubmed* citation network datasets [SNB+08]. We observe the following trends: First, as



(a) 2-layer networks with $(p, q) = (0.2, 0.02)$. (b) 3-layer networks with $(p, q) = (0.2, 0.02)$.



(c) 2-layer networks with $(p, q) = (0.02, 0.2)$. (d) 3-layer networks with $(p, q) = (0.02, 0.2)$.



(e) 2-layer networks with $(p, q) = (0.5, 0.1)$. (f) 3-layer networks with $(p, q) = (0.5, 0.1)$.

Figure 3.6: Averaged accuracy over 50 trials for networks with and without graph convolutions on the XOR-CSBM data model with $n = 400, d = 4$ and $\sigma = \frac{1}{\sqrt{d}}$. The x-axis denotes the ratio $\|\boldsymbol{\mu} - \boldsymbol{\nu}\|_2 / \sigma$ on a logarithmic scale. The vertical lines indicate the theoretical classification thresholds mentioned in Theorem 3 (red) and Theorem 4 (violet and pink).

claimed in Theorem 4, networks that utilize the graph perform remarkably better than a traditional MLP that does not use relational information. Second, all networks with one graph convolution in any layer achieve a mutually similar performance, and all networks with two graph convolutions in any combination of placement achieve a mutually similar performance. This demonstrates a result similar to Corollary 4.1 for real-world data. Finally, networks with two graph convolutions perform better than networks with one graph convolution, i.e., they obtain better accuracy on the same dataset.

In Fig. 3.7, we present for all networks, the maximum accuracy over 50 trials, where each trial corresponds to a random initialization of the networks. For 2-layer networks, the hidden layer has width 16, and for 3-layer networks, both hidden layers have width 16. We use a dropout probability of 0.5 and a weight decay of 10^{-5} while training.

For this study, we attribute minor changes in the accuracy to hyperparameters involving dropout and weight decay. This helps us clearly observe the important difference in the accuracy of networks with one graph convolution versus two graph convolutions. For example, in Fig. 3.7a, we note that there are differences in the accuracy of the networks with one graph convolution (red and blue). However, these differences are minor compared to the networks with one convolution (red and blue) and networks with two convolutions (green and yellow). Note that the accuracy slightly differs from well-known results in the literature due to implementation differences. In particular, the GCN implementation in [KW17] uses $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ as the normalized adjacency matrix, however, we use $\tilde{\mathbf{A}} = \mathbf{D}^{-1}\mathbf{A}$. (The proofs rely on degree concentration, and thus, generalize to the other type of normalization as well.) In Figs. 3.8 and 3.9, I show similar results for larger datasets, namely, OGBN-arXiv and OGBN-products [HFZ+20b].

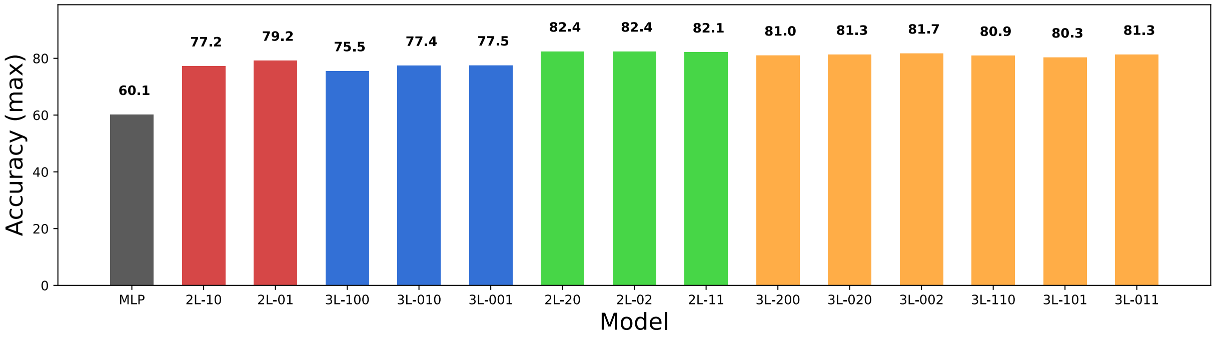
3.6 Additional Insights

In this section, we will look at some insightful results that are not directly relevant to the main results, yet, provide a meaningful chain of reasoning for a few of the research decisions in this thesis.

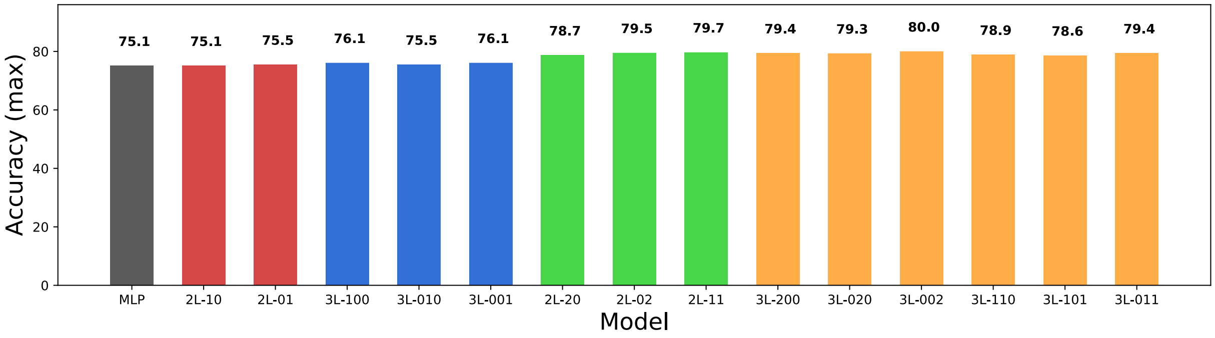
3.6.1 Variance Reduction Through Graph Convolutions

A natural question about graph convolutions is: What is the precise variance reduction obtained through relational information when a graph convolution operation is applied to the node attribute data? The following result answers this question under some assumptions.

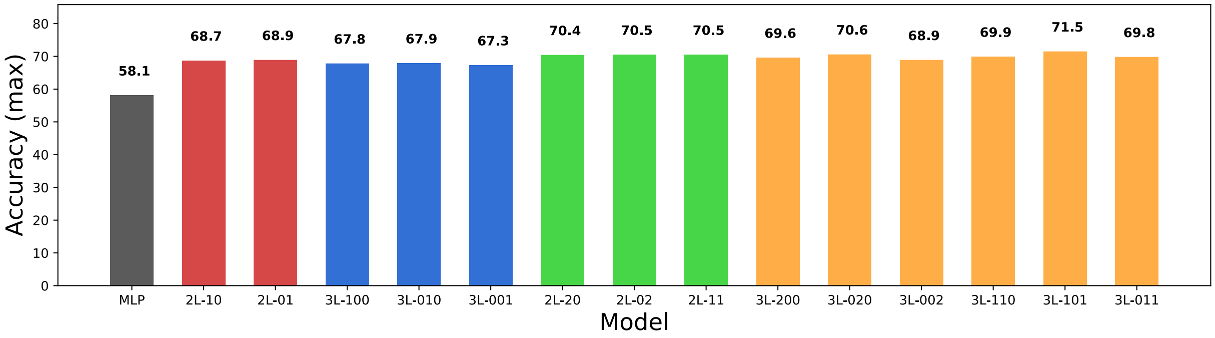
Lemma 3.6.1 (Variance reduction)**.** *Denote the event for Proposition 2.4.6 to be B. Let $\{\mathbf{X}_u\}_{u\in[n]} \in \mathbb{R}^{n\times d}$ be an iid sample of data. For a graph with adjacency matrix $\mathbf{A}$ (including*



(a) Accuracy of various learning models on the CORA dataset.

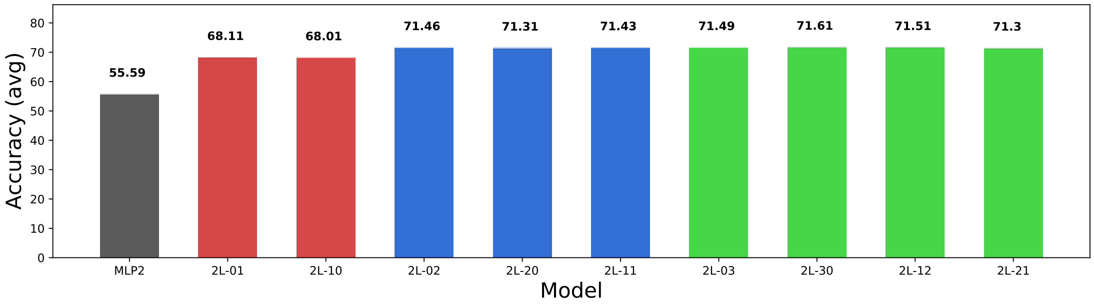


(b) Accuracy of various learning models on the Pubmed dataset.

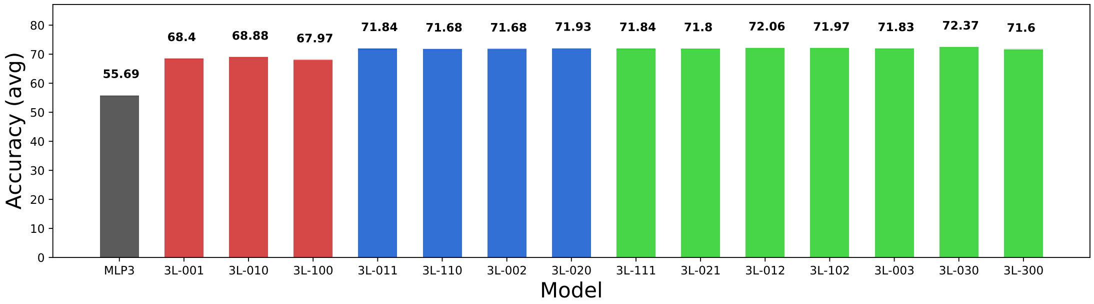


(c) Accuracy of various learning models on the CiteSeer dataset.

Figure 3.7: Maximum accuracy (percentage) over 50 trials. A network with k layers and $j_1, \ldots, j_k$ convolutions in each of the layers is represented by the label kL-$j_1 \ldots j_k$.



(a) OGBN-arXiv with two-layer networks.



(b) OGBN-arXiv with three-layer networks.

Figure 3.8: Average accuracy for OGB arXiv dataset over 10 trials. All models with one GC (red) perform mutually similarly, while models with two GCs (blue) and three GCs (green) perform mutually similarly and better than one GC.

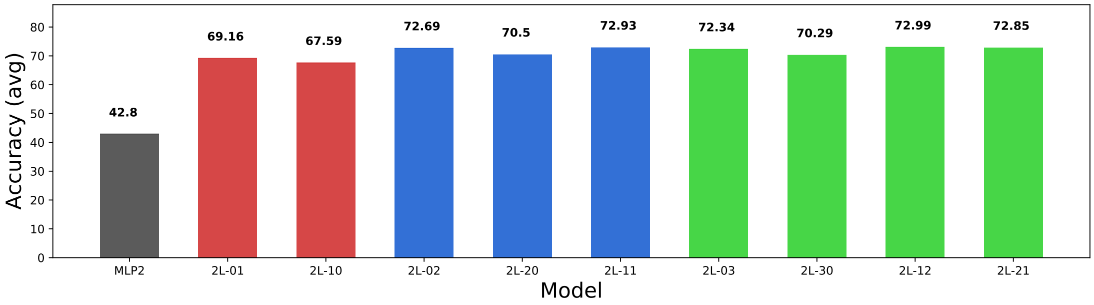
self-loops) and a fixed integer $K > 0$, define a K-convolution to be $\tilde{\mathbf{X}} = (\mathbf{D}^{-1}\mathbf{A})^K\mathbf{X}$. Then we have

$$\mathbf{Var}(\tilde{\mathbf{X}}_u \mid B) = \rho(K)\mathbf{Var}(\mathbf{X}_u), \ \textit{where } \rho(K) = \left(\frac{1+o_n(1)}{\Delta}\right)^{2K} \sum_{v\in[n]} \mathbf{A}^K(u,v)^2.$$

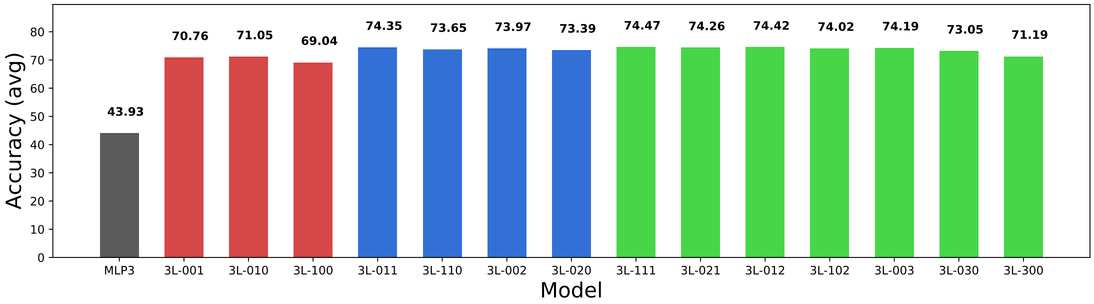
Here, $\mathbf{A}^K(u,v)$ is the entry in the uth row and vth column of the exponentiated matrix $\mathbf{A}^K$ and $\Delta = \mathbb{E}\,\mathbf{deg} = \frac{n}{2}(p+q)$.

Proof. For a matrix $\mathbf{M}$, the uth convolved data point is $\tilde{\mathbf{X}}_u = \mathbf{M}_u^\top\mathbf{X}$, where $\mathbf{M}_u^\top$ denotes the uth row of $\mathbf{M}$. Since $\mathbf{X}_u$ are iid, we have

$$\mathbf{Var}(\tilde{\mathbf{X}}_u) = \sum_{v\in[n]} (\mathbf{M}_{uv})^2\mathbf{Var}(\mathbf{X}_v).$$



(a) OGBN-products with two-layer networks.



(b) OGBN-products with three-layer networks.

Figure 3.9: Average accuracy for OGB products dataset over 10 trials. All models with one GC (red) perform mutually similarly, while models with two GCs (blue) and three GCs (green) perform mutually similarly and better than one GC.

It remains to compute the entries of the matrix $\mathbf{M} = (\mathbf{D}^{-1}\mathbf{A})^K$. Note that we have $\mathbf{D}^{-1}\mathbf{A}(u,v) = {}^{a_{uv}}/_{\mathbf{deg}(u)}$, so we obtain that

$$\mathbf{M}_{uv} = (\mathbf{D}^{-1}\mathbf{A})^K(u,v) = \sum_{w_1=1}^{n}\sum_{w_2=1}^{n}\cdots\sum_{w_{K-1}=1}^{n} \frac{a_{uw_1}a_{w_1w_2}\cdots a_{w_{K-2}w_{K-1}}a_{w_{K-1}v}}{\mathbf{deg}(u)\mathbf{deg}(w_1)\cdots\mathbf{deg}(w_{K-1})}.$$

Recall that on the event B, the degrees of all nodes are $\Delta(1 \pm o_n(1))$, and hence, we have

$$\mathbf{M}_{uv} = \frac{(1 \pm o_n(1))^K}{\Delta^K} \sum_{w_1=1}^{n}\sum_{w_2=1}^{n}\cdots\sum_{w_{K-1}=1}^{n} a_{uw_1}\cdots a_{w_{K-2}w_{K-1}}a_{w_{K-1}v},$$

where the error $o_n(1) = O(\frac{1}{\sqrt{\log n}})$. The sum of these products of the entries of $\mathbf{A}$ is simply the number of length-K paths from node i to j, i.e., $\mathbf{A}^K(i,j)$. Thus, we have

$$\mathbf{Var}(\tilde{\mathbf{X}}_u \mid B) = \sum_{j\in[n]} (\mathbf{M}_{uv})^2 \mathbf{Var}(\mathbf{X}_v) = \left(\frac{1+o_n(1)}{\Delta}\right)^{2K} \sum_{j\in[n]} \mathbf{A}^K(i,j)^2 \mathbf{Var}(\mathbf{X}_v).$$

Since $\mathbf{X}_v$ are iid, we obtain that $\rho(K) = \left(\frac{1+o_n(1)}{\Delta}\right)^{2K} \sum_{j\in[n]} \mathbf{A}^K(i,j)^2$. $\square$

Let us briefly discuss the implications of Lemma 3.6.1. Consider a sample $(\mathbf{A},\mathbf{X})$ drawn from XOR-CSBM$(n,p,q,\boldsymbol{\mu},\boldsymbol{\nu},\sigma)$ for the symmetric case where exactly $n/2$ nodes are in each of the two classes. We have that

$$\mathbb{E}\mathbf{A} = \begin{pmatrix} p\mathbf{I}_{n/2} & q\mathbf{I}_{n/2} \\ q\mathbf{I}_{n/2} & p\mathbf{I}_{n/2} \end{pmatrix}.$$

This gives us $\mathbb{E}\rho(K) \approx \frac{1}{n}(1+\gamma(p,q)^{2K})$ for any $K \geq 2$. Recall that a single graph convolution reduces the distance between the means by a factor of $\gamma(p,q)$. Hence, to comment on the performance of an arbitrary number of convolutions, K, we might hope to compare the reduction in this distance, $\gamma(p,q)^K$ with the reduction in the variance, $\rho(K)$ to obtain a condition on K in terms of n, p, and q. The challenge, however, lies in the fact that in deeper layers, computing $\rho(K)$ is non-trivial due to node features being highly correlated. Moreover, an argument to claim that $\rho(K) \approx \mathbb{E}\rho(K)$ is needed for this approach, which seems to require strong density assumptions on the graph.

3.6.2 Thresholds for a Simpler Case

Although Theorem 4 encapsulates the general condition for networks with up to two graph convolutions to achieve perfect classification, let us understand the theorem in a simplified setting where $\gamma(p,q) = \Omega(1)$. In this regime, one can analyze two cases:

1. Case $\beta = \Omega(\sigma)$: Using part two of Lemma 3.3.3 implies that $\zeta = \Omega(\frac{\beta}{\sigma})$. Hence, for one graph convolution, the condition $\gamma(p,q)\zeta = \omega\left(\sqrt{\frac{\log n}{n(p+q)}}\right)$ is satisfied when $\frac{\beta}{\sigma} = \omega\left(\sqrt{\frac{\log n}{n(p+q)}}\right)$, implying that $\beta = \omega\left(\sigma\sqrt{\frac{\log n}{n(p+q)}}\right)$. Similarly, for two graph convolutions, the condition $\gamma(p,q)^2\zeta = \omega\left(\sqrt{\frac{\log n}{n}}\right)$ is satisfied when $\beta = \omega\left(\sigma\sqrt{\frac{\log n}{n}}\right)$.

2. Case $\beta = o(\sigma)$: Using part three of Lemma 3.3.3 implies that $\zeta = \Omega(\frac{\beta^2}{\sigma^2})$. Hence, for one graph convolution, the condition $\gamma(p,q)\zeta = \omega\left(\sqrt{\frac{\log n}{n(p+q)}}\right)$ is satisfied when $\left(\frac{\beta}{\sigma}\right)^2 = \omega\left(\sqrt{\frac{\log n}{n(p+q)}}\right)$, implying that $\beta = \omega\left(\sigma\sqrt[4]{\frac{\log n}{n(p+q)}}\right)$. Similarly, for two graph convolutions, the condition $\gamma(p,q)^2\zeta = \omega\left(\sqrt{\frac{\log n}{n}}\right)$ is satisfied when $\beta = \omega\left(\sigma\sqrt[4]{\frac{\log n}{n}}\right)$.

Combining both cases, we find that the theorems imply perfect classification if:

$$\beta = \begin{cases} \Omega\left(\frac{\sigma\sqrt{\log n}}{\sqrt[4]{n(p+q)}}\right) & \text{for networks wth one graph convolution,} \\ \Omega\left(\frac{\sigma\sqrt{\log n}}{\sqrt[4]{n}}\right) & \text{for networks with two graph convolutions.} \end{cases}$$

3.6.3 Graph Convolution in the First Layer

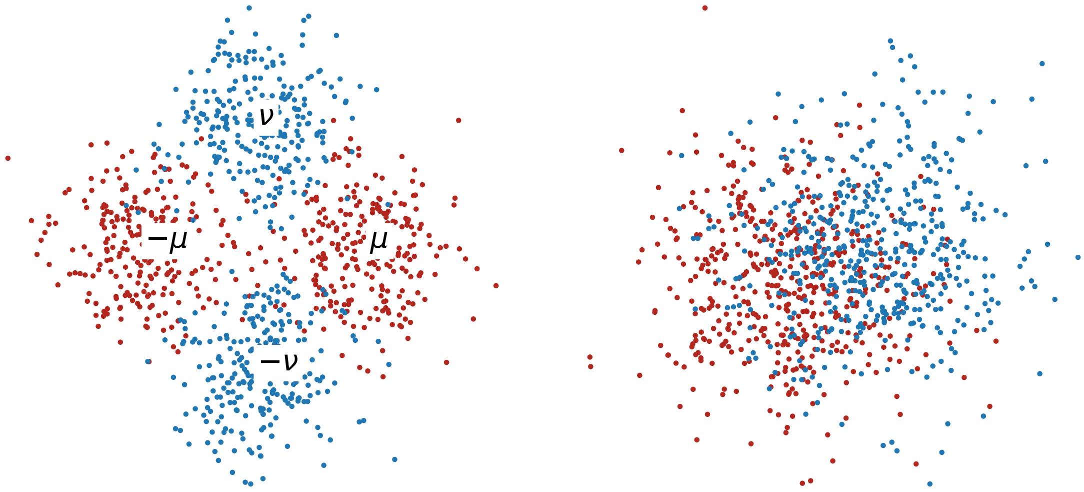
In this section, we show precisely why a graph convolution operation in the first layer is detrimental to the classification task. This effect is visualized in Fig. 3.10, and is attributed to the averaging of data points in the same class but different components of the mixture that have means with opposite signs. As n (the sample size) grows, the difference between the averages of node features over the two classes diminishes (see Figs. 3.10a and 3.10b). In other words, the means of the two classes collapse to the same point for large n. However, in the last layer, since the input consists of transformed features that are linearly separable, a graph convolution helps with the classification task (see Figs. 3.10c and 3.10d).

In Fig. 3.11, we empirically show that placing a graph convolution in the first layer makes the classification task difficult since the means of the convolved data collapse towards 0.

Proposition 3.6.2. *Fix a positive integer $d > 0$, $\sigma \in \mathbb{R}^+$ and $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^d$. Let $(\mathbf{A}, \mathbf{X}) \sim$ XOR-CSBM$(n, p, q, \boldsymbol{\mu}, \boldsymbol{\nu}, \sigma)$. Define $\tilde{\mathbf{X}}$ to be the transformed data after applying a graph convolution on $\mathbf{X}$, i.e., $\tilde{\mathbf{X}} = \mathbf{D}^{-1}\mathbf{A}\mathbf{X}$. Then in the regime where $p, q = \Omega(\frac{\log^2 n}{n})$, with probability at least $1 - 1/\mathrm{poly}(n)$ we have that*

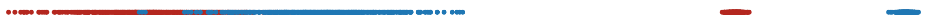
$$\mathbb{E}\tilde{\mathbf{X}}_i = \begin{cases} \dfrac{p\boldsymbol{\mu} + q\boldsymbol{\nu}}{2(p+q)} \cdot o_n(1) & i \in C_0 \\ \dfrac{p\boldsymbol{\nu} + q\boldsymbol{\mu}}{2(p+q)} \cdot o_n(1) & i \in C_1 \end{cases}.$$

Hence, the distance between the means of the convolved data, given by $\frac{p-q}{2(p+q)}\beta \cdot o_n(1)$ diminishes to 0 for $n \to \infty$.



(a) Original node features at the first layer.

(b) After GC at the first layer.



(c) Feature representation at the last layer.

(d) After GC at the last layer.

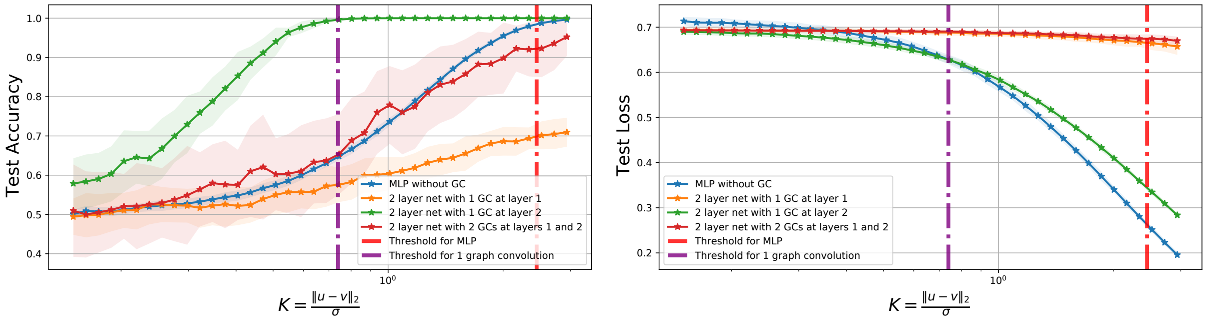
Figure 3.10: Placement of a graph convolution (GC) in the first layer versus the last layer for data sampled from the XOR-CSBM. For this figure we used 1000 nodes in each class and a randomly sampled stochastic block-model graph with $p = 0.8$ and $q = 0.2$.

Proof. Fix $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^d$ and define the following sets:

$$C_{-\boldsymbol{\mu}} = \{i \mid y_i = 0, \eta_i = 0\}, \qquad C_{\boldsymbol{\mu}} = \{i \mid y_i = 0, \eta_i = 1\},$$
$$C_{-\boldsymbol{\nu}} = \{i \mid y_i = 1, \eta_i = 0\}, \qquad C_{\boldsymbol{\nu}} = \{i \mid y_i = 1, \eta_i = 1\}.$$

Denote $\tilde{\mathbf{X}} = \mathbf{D}^{-1}\mathbf{A}\mathbf{X}$ and note that for any $i \in [n]$, the row vector

$$\begin{aligned}\tilde{\mathbf{X}}_i &= \frac{1}{\mathbf{deg}(i)} \sum_{j\in[n]} a_{ij}\mathbf{X}_j = \frac{1}{\mathbf{deg}(i)} \sum_{j\in[n]} a_{ij}(\mathbb{E}\mathbf{X}_j + \sigma\mathbf{g}_j) \\ &= \frac{1}{\mathbf{deg}(i)} \left[\boldsymbol{\mu}\left(\sum_{j\in C_{\boldsymbol{\mu}}} a_{ij} - \sum_{j\in C_{-\boldsymbol{\mu}}} a_{ij}\right) + \boldsymbol{\nu}\left(\sum_{j\in C_{\boldsymbol{\nu}}} a_{ij} - \sum_{j\in C_{-\boldsymbol{\nu}}} a_{ij}\right) + \sigma \sum_{j\in[n]} a_{ij}\mathbf{g}_j\right],\end{aligned}$$



(a) Test accuracy.

(b) Test loss.

Figure 3.11: Comparing the accuracy and loss for various networks with and without graph convolutions, averaged over 50 trials. Networks with a graph convolution in the first layer (red and orange) fail to generalize even for a large distance between the means of the data. For this experiment, we set $n = 400$ and $d = 4$, with $\sigma^2 = 1/d$.

where we used the fact that $\mathbf{X}_j = (2\eta_j - 1)((1-y_j)\boldsymbol{\mu} + y_j\boldsymbol{\nu} + \sigma\mathbf{g}_j)$ for a set of iid Gaussian random vectors $\mathbf{g}_j \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$.

Note that since ϵ_i, η_i are Bernoulli random variables, using the Chernoff bound [Ver18, Section 2], we have that with probability at least $1 - 1/\text{poly}(n)$,

$$|C_{-\boldsymbol{\mu}}| = |C_{\boldsymbol{\mu}}| = |C_{-\boldsymbol{\nu}}| = |C_{\boldsymbol{\nu}}| = \frac{n}{4}(1 \pm o_n(1)).$$

We now use an argument similar to how we obtained Proposition 2.4.6 to observe that for any $c > 0$, with probability at least $1 - O(n^{-c})$, the following holds for all $i \in [n]$:

$$\frac{1}{\mathbf{deg}(i)}\left(\sum_{j\in C_{\boldsymbol{\mu}}} a_{ij} - \sum_{j\in C_{-\boldsymbol{\mu}}} a_{ij}\right) = O\left(\frac{(1-y_i)p + y_i q}{2(p+q)}\sqrt{\frac{c}{\log n}}\right),$$

$$\frac{1}{\mathbf{deg}(i)}\left(\sum_{j\in C_{\boldsymbol{\nu}}} a_{ij} - \sum_{j\in C_{-\boldsymbol{\nu}}} a_{ij}\right) = O\left(\frac{y_i p + (1-y_i) q}{2(p+q)}\sqrt{\frac{c}{\log n}}\right).$$

Hence, we have that for all $i \in [n]$,

$$\mathbb{E}\tilde{\mathbf{X}}_i = \left[\left(\frac{(1-y_i)p + y_i q}{2(p+q)}\right)\boldsymbol{\mu} + \left(\frac{y_i p + (1-y_i)q}{2(p+q)}\right)\boldsymbol{\nu}\right] \cdot O\left(\sqrt{\frac{c}{\log n}}\right)$$

$$= \begin{cases} \dfrac{p\boldsymbol{\mu} + q\boldsymbol{\nu}}{2(p+q)} \cdot o_n(1) & i \in C_0 \\ \dfrac{p\boldsymbol{\nu} + q\boldsymbol{\mu}}{2(p+q)} \cdot o_n(1) & i \in C_1 \end{cases}$$

Using the above result, we obtain that for fixed $\boldsymbol{\mu}, \boldsymbol{\nu}$ and σ, the distance between the means after convolution diminishes to 0 as $n \to \infty$. $\square$

Chapter 4

Analysis of Graph Attention

4.1 Preliminaries

This section describes the *Graph Attention* mechanism [VCC$^+$18a] along with notations and terms that are frequently used throughout this chapter.

Define two classes as $C_b = \{u \in [n] \mid y_u = b\}$ for $b \in \{0,1\}$. For each index $u \in [n]$, set the feature vector $\mathbf{X}_u \in \mathbb{R}^d$ as $\mathbf{X}_i \sim \mathcal{N}((2y_i - 1)\boldsymbol{\mu}, \sigma^2\mathbf{I})$, where $\boldsymbol{\mu} \in \mathbb{R}^d$. This is a binary CSBM with means $\pm\boldsymbol{\mu}$. The results can be easily generalized to general means, however, we will look at the symmetric case here. For a given pair $p, q \in [0,1]$, we look at the graph $G = (\mathbf{A}, \mathbf{X}) \sim \mathrm{CSBM}(n, p, q, \boldsymbol{\mu}, \sigma)$.

An advantage of CSBM is that it allows us to control the noise by controlling the parameters of the distributions of the model. In particular, CSBM allows us to control the distance between the means and the variance of the Gaussians, which are important for controlling the separability. For example, fixing the variance, we see that the closer the means are, the more difficult the separation of the node features becomes. Another notable advantage of CSBM is that it allows us to control the structural noise and homophily level of the graph. The level of noise in the graph is controlled by increasing or decreasing the gap between the intra-class edge probability p and the inter-class edge probability q, and the level of homophily in the graph is controlled by the relative magnitude between p and q. For example, when p is much larger than q, then a node is more likely to be connected with a node from the same class, and hence we obtain a homophilous graph; when q is much larger than p, then a node is more likely to be connected with a node from a different class, and hence we obtain a heterophilous graph. There are several recent works exploring the behaviour of GNNs in heterophilous graphs [LHL+21a, YHS+22, BDGC+22, LHL+22]. Interestingly, the results presented in this thesis for graph attention's behaviour over the CSBM data model do not depend on whether the graph is homophilous or heterophilous.

Given node representations $\boldsymbol{h}_v \in \mathbb{R}^{F'}$ for $v \in [n]$, a (spatial) graph convolution for node u produces the output $\boldsymbol{h}'_u$ as follows:

$$\boldsymbol{h}'_u = \sum_{v \in [n]} c_{uv} a_{uv} \mathbf{W} \boldsymbol{h}_v, \quad c_{uv} = \frac{1}{\mathbf{deg}(u)},$$

where $\mathbf{W} \in \mathbb{R}^{F \times F'}$ is a learnable matrix. Throughout this chapter, we refer to this operation as *simple graph convolution* or *standard graph convolution*. Our definition of graph convolution is essentially the mean aggregation step in a general GNN layer [HYL17]. The normalization constant c_{uv} in our definition is closely related to the symmetric normalization $c_{uv} = (\mathbf{deg}(u))^{-1/2}(\mathbf{deg}(v))^{-1/2}$ used in the original Graph Convolutional Network (GCN) introduced by [KW17]. Our definition does not affect the discussions or implications we have for GCN with symmetric normalization due to degree concentration. More broadly, there are other forms of graph convolutions in the literature [BBCV21, DBV16, LMBB18], which are not compared within this work.

A *single-head* graph attention applies some weight function on the edges based on their node features (or a mapping thereof). Given two representations $\boldsymbol{h}_u, \boldsymbol{h}_v \in \mathbb{R}^{F'}$ for two nodes $u, v \in [n]$, the *attention model/mechanism* is defined as the mapping

$$\Psi(\boldsymbol{h}_u, \boldsymbol{h}_v) \overset{\text{def}}{=} \alpha(\mathbf{W}\boldsymbol{h}_u, \mathbf{W}\boldsymbol{h}_v)$$

where $\alpha : \mathbb{R}^F \times \mathbb{R}^F \to \mathbb{R}$ and $\mathbf{W} \in \mathbb{R}^{F \times F'}$ is a learnable matrix. The *attention coefficient* for a node u and its neighbour v is defined as

$$\tau_{uv} \overset{\text{def}}{=} \frac{\exp(\Psi(\boldsymbol{h}_u, \boldsymbol{h}_v))}{\sum_{w \in \eta_1(u)} \exp(\Psi(\boldsymbol{h}_u, \boldsymbol{h}_w))}, \tag{4.1}$$

where $\eta_1(u)$ is the set of neighbours of node u that also includes node u itself. Let f be some element-wise activation function (which is usually nonlinear), the graph attention convolution output for a node $u \in [n]$ is given by

$$\begin{aligned} \boldsymbol{h}'_u &= \sum_{v \in [n]} \mathbf{A}_{uv} \tau_{uv} \mathbf{W} \boldsymbol{h}_v, \\ \tilde{\boldsymbol{h}}_u &= f(\boldsymbol{h}'_u). \end{aligned} \tag{4.2}$$

A *multi-head* graph attention [VCC+18a] uses $K \in \mathbb{N}$ weight matrices $\mathbf{W}^1, \ldots, \mathbf{W}^K \in \mathbb{R}^{F \times F'}$ and averages their individual (single-head) outputs. We consider the most simplified case of a single graph attention layer (i.e., $F' = d$ and $F = 1$) where α is realized by an MLP using the LeakyRelu activation function. The LeakyRelu activation function is defined as $\mathrm{LeakyRelu}(x) = x$ if $x \geq 0$ and $\mathrm{LeakyRelu}(x) = \beta x$ for some constant $\beta \in [0, 1)$ if $x < 0$.

A natural requirement of attention architectures is to maintain important edges in the graph and ignore unimportant edges. For example, important edges could be the set of intra-class edges and unimportant edges could be the set of inter-class edges. In this case, if graph attention maintains all intra-class edges and ignores all inter-class edges, then a node from a class will be connected only to nodes from the same class. More specifically, a node v will be connected to neighbour nodes whose associated node features come from the *same distribution* as node features of v. Given two sets A and B, we denote $A \times B \overset{\text{def}}{=} \{(i, j) : i \in A, j \in B\}$ and $A^2 \overset{\text{def}}{=} A \times A$.

4.2 Edge Separation in the Hard Regime

In this regime ($\|\boldsymbol{\mu}\| = \kappa\sigma$ for $\kappa \leq O(\sqrt{\log n})$), we show that *every* attention architecture Ψ fails to separate the edges if $\kappa < \sqrt{2 \log n}$. The goal of the attention mechanism is to decide whether an edge (u, v) is an inter-class edge or an intra-class edge based on the node feature vectors $\mathbf{X}_u$ and $\mathbf{X}_v$. Let $\mathbf{X}'_{uv}$ denote the vector obtained from concatenating $\mathbf{X}_u$ and $\mathbf{X}_v$, that is,

$$\mathbf{X}'_{uv} \overset{\text{def}}{=} \begin{pmatrix} \mathbf{X}_u \\ \mathbf{X}_v \end{pmatrix}. \tag{4.3}$$

We want to analyze every classifier h' which takes as input $\mathbf{X}'_{uv}$ and tries to separate inter-class edges and intra-class edges. An ideal classifier would have the property

$$y = h'(\mathbf{X}'_{uv}) = \begin{cases} 0, & \text{if } (u, v) \text{ is an inter-class edge}, \\ 1, & \text{if } (u, v) \text{ is an intra-class edge}. \end{cases} \tag{4.4}$$

To understand the limitations of all such classifiers in this regime, it suffices to consider the Bayes optimal classifier for this data model, whose probability of misclassifying of an arbitrary edge lower bounds that of every attention architecture which takes as input $(\mathbf{X}_u, \mathbf{X}_v)$. Consequently, by deriving a misclassification rate for the Bayes classifier, we obtain a lower bound on the misclassification rate for every attention mechanism Ψ for classifying intra-class and inter-class edges. The following Lemma 4.2.1 describes the Bayes classifier for this classification task.

Lemma 4.2.1. *Let* $(\mathbf{A}, \mathbf{X}) \sim \mathrm{CSBM}(n, p, q, \boldsymbol{\mu}, \sigma)$ *and let* $\mathbf{X}'_{uv}$ *be defined as in* (4.3)*. The Bayes optimal classifier for* $\mathbf{X}'_{uv}$ *is realized by the following function,*

$$h^*(\boldsymbol{x}) = \begin{cases} 0, & \textit{if } p \cosh\left(\frac{\boldsymbol{x}^\top \boldsymbol{\mu}'}{\sigma^2}\right) \le q \cosh\left(\frac{\boldsymbol{x}^\top \boldsymbol{\nu}'}{\sigma^2}\right), \\ 1, & \textit{otherwise}, \end{cases} \tag{4.5}$$

where $\boldsymbol{\mu}' \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu} \end{pmatrix}$ *and* $\boldsymbol{\nu}' \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{\mu} \\ -\boldsymbol{\mu} \end{pmatrix}$.

Proof. Note that $\mathbf{X}'_{uv}$ is a mixture of $2d$-dimensional Gaussian distributions,

$$\mathbf{X}'_{uv} \sim \begin{cases} \mathcal{N}(-\boldsymbol{\mu}', \sigma^2 \mathbf{I}) & u \in C_0, v \in C_0 \\ \mathcal{N}(\boldsymbol{\mu}', \sigma^2 \mathbf{I}) & u \in C_1, v \in C_1 \\ \mathcal{N}(-\boldsymbol{\nu}', \sigma^2 \mathbf{I}) & u \in C_0, v \in C_1 \\ \mathcal{N}(\boldsymbol{\nu}', \sigma^2 \mathbf{I}) & u \in C_1, v \in C_0 \end{cases}.$$

The optimal classifier is then given by

$$h^*(\boldsymbol{x}) = \underset{c \in \{0,1\}}{\operatorname{argmax}} \, \mathbf{Pr}\left[y = c \mid \boldsymbol{x}\right].$$

Note that $\mathbf{Pr}\left[y = 0\right] = \frac{q}{p+q}$ and $\mathbf{Pr}\left[y = 1\right] = \frac{p}{p+q}$. Thus, by Bayes rule, we obtain that

$$\begin{aligned} \mathbf{Pr}\left[y = c \mid \boldsymbol{x}\right] &= \frac{\mathbf{Pr}\left[y = c\right] \cdot f_{\boldsymbol{x}|y}(\boldsymbol{x} \mid y = c)}{\mathbf{Pr}\left[y = 0\right] f_{\boldsymbol{x}|y=0}(\boldsymbol{x} \mid y = 0) + \mathbf{Pr}\left[y = 1\right] f_{\boldsymbol{x}|y=1}(\boldsymbol{x} \mid y = 1)} \\ &= \frac{1}{1 + \frac{\mathbf{Pr}[y=1-c] \cdot f_{\boldsymbol{x}|y}(\boldsymbol{x}|y=1-c)}{\mathbf{Pr}[y=c] \cdot f_{\boldsymbol{x}|y}(\boldsymbol{x}|y=c)}}. \end{aligned}$$

Suppose that $\boldsymbol{x} = \mathbf{X}'_{uv}$ such that $y_u \neq y_v$. Then $h^*(\boldsymbol{x}) = 0$ if and only if $\mathbf{Pr}\left[y = 0 \mid \boldsymbol{x}\right] \ge \frac{1}{2}$. Hence, for $c = 0$ we require that

$$\frac{\mathbf{Pr}\left[y = 1 - c\right] \cdot f_{\boldsymbol{x}|y}(\boldsymbol{x} \mid y = 1 - c)}{\mathbf{Pr}\left[y = c\right] \cdot f_{\boldsymbol{x}|y}(\boldsymbol{x} \mid y = c)} = \frac{p}{q} \frac{f_{\boldsymbol{x}|y}(\boldsymbol{x} \mid y = 1)}{f_{\boldsymbol{x}|y}(\boldsymbol{x} \mid y = 0)} = \frac{p}{q} \frac{\cosh\left(\frac{1}{\sigma^2} \boldsymbol{x}^\top \boldsymbol{\mu}'\right)}{\cosh\left(\frac{1}{\sigma^2} \boldsymbol{x}^\top \boldsymbol{\nu}'\right)} \le 1,$$

Similarly, we obtain the reverse condition for $h^*(\boldsymbol{x}) = 1$. □

Using Lemma 4.2.1, we can lower bound the rate of misclassification of edges that every attention mechanism Ψ exhibits. Below we define $\Phi_{\mathrm{c}} \stackrel{\text{def}}{=} 1 - \Phi$, where Φ is the cumulative distribution function of $\mathcal{N}(0, 1)$.

Theorem 5. *Suppose* $\|\boldsymbol{\mu}\| = \kappa\sigma$ *for some* $\kappa > 0$ *and let* Ψ *be any attention mechanism. Then with probability* $1 - o_n(1)$*, we have that,*

1. Ψ *fails to correctly classify at least* $2 \cdot \Phi_{\mathrm{c}}(\kappa)^2$ *fraction of inter-class edges;*
2. *For any* $K > 0$ *if* $q > \frac{K \log^2 n}{n\Phi_{\mathrm{c}}(\kappa)^2}$*, then with probability at least* $1 - O(n^{-8K\Phi_{\mathrm{c}}(\kappa)^2 \log n})$*,* Ψ *misclassifies at least one inter-class edge.*

Part 1 of Theorem 5 implies that if $\|\boldsymbol{\mu}\|$ is *linear* in the standard deviation σ, that is if $\kappa = O(1)$, then with overwhelming probability the attention mechanism fails to distinguish a constant fraction of inter-class edges from intra-class edges. Furthermore, part 2 of Theorem 5 characterizes a regime for the inter-class edge probability q where the attention mechanism fails to distinguish at least one inter-class edge. It provides a lower bound on q in terms of the scale at which the distance between the means grows compared to the standard deviation σ. This aligns with the intuition that increasing the distance between the means makes it easier for the attention mechanism to correctly distinguish inter-class and intra-class edges. However, if q is also increased with the right proportion, in other words, if the noise in the graph is increased, then the attention mechanism would still fail to correctly distinguish at least one inter-class edge. For instance, for $\kappa = \sqrt{2 \log\log n}$ and $K = \log^2 n$, we get that if $q > \Omega(\frac{\log^{6+o(1)} n}{n})$, then with probability at least $1 - o(1)$, Ψ misclassifies at least one inter-class edge.

The proof of Theorem 5 relies on analyzing the behaviour of the Bayes optimal classifier in (4.5). We compute an upper bound on the probability with which the optimal classifier correctly classifies an arbitrary inter-class edge. Then the proof of part 1 of Theorem 5 follows from a concentration argument for the fraction of inter-class edges that are misclassified by the optimal classifier. For part 2, we use a similar concentration argument to choose a suitable threshold for q that forces the optimal classifier to fail on at least one inter-class edge. The formal arguments are provided in the next section.

4.3 Proof of Theorem 5

Proof. From Lemma 4.2.1, we observe that for successful classification by the optimal classifier, we need

$$
\begin{aligned}
p\cosh\left(\frac{\boldsymbol{x}^\top \boldsymbol{\mu}'}{\sigma^2}\right) &\le q\cosh\left(\frac{\boldsymbol{x}^\top \boldsymbol{\nu}'}{\sigma^2}\right) \quad \text{for } y_i \neq y_j,\\
p\cosh\left(\frac{\boldsymbol{x}^\top \boldsymbol{\mu}'}{\sigma^2}\right) &> q\cosh\left(\frac{\boldsymbol{x}^\top \boldsymbol{\nu}'}{\sigma^2}\right) \quad \text{for } y_i = y_j.
\end{aligned}
$$

We will split the analysis into two cases. First, note that when $p \geq q$ we have for $y_i \neq y_j$ that

$$p \cosh\left(\tfrac{\boldsymbol{x}^\top \boldsymbol{\mu}'}{\sigma^2}\right) \leq q \cosh\left(\tfrac{\boldsymbol{x}^\top \boldsymbol{\nu}'}{\sigma^2}\right) \implies \cosh\left(\tfrac{\boldsymbol{x}^\top \boldsymbol{\mu}'}{\sigma^2}\right) \leq \cosh\left(\tfrac{\boldsymbol{x}^\top \boldsymbol{\nu}'}{\sigma^2}\right) \implies |\boldsymbol{x}^\top \boldsymbol{\mu}'| \leq |\boldsymbol{x}^\top \boldsymbol{\nu}'|.$$

In the first implication, we used that $p \geq q$, while the second implication follows from the fact that $\cosh(a) \leq \cosh(b) \implies |a| \leq |b|$ for all $a, b \in \mathbb{R}$. Similarly, for $p < q$ we have for $y_i = y_j$ that

$$p \cosh\left(\tfrac{\boldsymbol{x}^\top \boldsymbol{\mu}'}{\sigma^2}\right) > q \cosh\left(\tfrac{\boldsymbol{x}^\top \boldsymbol{\nu}'}{\sigma^2}\right) \implies \cosh\left(\tfrac{\boldsymbol{x}^\top \boldsymbol{\mu}'}{\sigma^2}\right) > \cosh\left(\tfrac{\boldsymbol{x}^\top \boldsymbol{\nu}'}{\sigma^2}\right) \implies |\boldsymbol{x}^\top \boldsymbol{\mu}'| > |\boldsymbol{x}^\top \boldsymbol{\nu}'|.$$

Therefore, for each of the above cases, we can upper bound the probability for either $y_i = y_j$ or $y_i \neq y_j$ that $\mathbf{X}'_{ij}$ is correctly classified, by the probability of the event $|\mathbf{X}'^T_{ij} \boldsymbol{\mu}'| \leq |\mathbf{X}'^T_{ij} \boldsymbol{\nu}'|$ or equivalently $|\mathbf{X}'^T_{ij} \boldsymbol{\mu}'| > |\mathbf{X}'^T_{ij} \boldsymbol{\nu}'|$. We focus on the former as the latter is equivalent and symmetric. Writing $\mathbf{X}_i = \boldsymbol{\mu} + \sigma \boldsymbol{g}_i$ and $\mathbf{X}_j = -\boldsymbol{\mu} + \sigma \boldsymbol{g}_j$, we have that for $i \in C_1$ and $j \in C_0$,

$$\begin{aligned}
\mathbf{Pr}\left[h^*(\mathbf{X}'_{ij}) = 0\right] &\leq \mathbf{Pr}\left[|\mathbf{X}'^T_{ij} \boldsymbol{\mu}'| \leq |\mathbf{X}'^T_{ij} \boldsymbol{\nu}'|\right] \\
&= \mathbf{Pr}\left[|\mathbf{X}_i^\top \boldsymbol{\mu} + \mathbf{X}_j^\top \boldsymbol{\mu}| \leq |\mathbf{X}_i^\top \boldsymbol{\mu} - \mathbf{X}_j^\top \boldsymbol{\mu}|\right] \\
&= \mathbf{Pr}\left[\sigma |\boldsymbol{g}_i^\top \boldsymbol{\mu} + \boldsymbol{g}_j^\top \boldsymbol{\mu}| \leq |\pm 2\|\boldsymbol{\mu}\|^2 + \sigma \boldsymbol{g}_i^\top \boldsymbol{\mu} - \sigma \boldsymbol{g}_j^\top \boldsymbol{\mu}|\right] \\
&\leq \mathbf{Pr}\left[|\boldsymbol{g}_i^\top \hat{\boldsymbol{\mu}} + \boldsymbol{g}_j^\top \hat{\boldsymbol{\mu}}| - |\boldsymbol{g}_i^\top \hat{\boldsymbol{\mu}} - \boldsymbol{g}_j^\top \hat{\boldsymbol{\mu}}| \leq \frac{2\|\boldsymbol{\mu}\|}{\sigma}\right] \\
&= \mathbf{Pr}\left[|\boldsymbol{g}_i^\top \hat{\boldsymbol{\mu}} + \boldsymbol{g}_j^\top \hat{\boldsymbol{\mu}}| - |\boldsymbol{g}_i^\top \hat{\boldsymbol{\mu}} - \boldsymbol{g}_j^\top \hat{\boldsymbol{\mu}}| \leq 2\kappa\right],
\end{aligned}$$

where we denote $\hat{\boldsymbol{\mu}} = \boldsymbol{\mu} / \|\boldsymbol{\mu}\|$. In the second to last step above, we used triangle inequality to pull $2\|\boldsymbol{\mu}\|^2$ outside the absolute value, while in the last equation, we use $\|\boldsymbol{\mu}\| = \kappa\sigma$.

Let $z_i = \boldsymbol{g}_i^\top \hat{\boldsymbol{\mu}}$ for all $i \in [n]$. Then the above probability is $\mathbf{Pr}\left[|z_i + z_j| - |z_i - z_j| \leq 2\kappa\right]$, where $z_i, z_j \sim \mathcal{N}(0, 1)$ are independent random variables. Note that we have

$$\begin{aligned}
\mathbf{Pr}\left[h^*(\mathbf{X}'_{ij}) = 0\right] &\leq \mathbf{Pr}\left[|z_i + z_j| - |z_i - z_j| \leq 2\kappa\right] \\
&= \mathbf{Pr}\left[|z_i + z_j| - |z_i - z_j| \leq 2\kappa, |z_i| \leq \kappa\right] \\
&\quad + \mathbf{Pr}\left[|z_i + z_j| - |z_i - z_j| \leq 2\kappa, |z_i| > \kappa\right] \\
&= \mathbf{Pr}\left[|z_i| \leq \kappa\right] + \Phi(\kappa) \mathbf{Pr}\left[|z_i| > \kappa\right]. \qquad (4.6)
\end{aligned}$$

To see how we obtain the last equation, observe that if $|z_i| \leq \kappa$ then we have

$$|z_i + z_j| - |z_i - z_j| = |z_i + z_j| - |z_j - z_i|$$

$$
\begin{aligned}
&\le |z_i| + |z_j| - |z_j - z_i| && \text{by triangle inequality}\\
&\le |z_i| + |z_j| - \big||z_j| - |z_i|\big| && \text{by reverse triangle inequality}\\
&\le |z_i| + |z_j| - (|z_j| - |z_i|) = 2|z_i|\\
&\le 2\kappa,
\end{aligned}
$$

hence, $\mathbf{Pr}\left[|z_i + z_j| - |z_i - z_j| \le 2\kappa, |z_i| \le \kappa\right] = \mathbf{Pr}\left[|z_i| \le \kappa\right]$. On the other hand, for $|z_i| > \kappa$, we look at each case, conditioned on the events $z_i > \kappa$ and $z_i < -\kappa$ for each of the four cases based on the signs of $z_i + z_j$ and $z_i - z_j$. We denote by E the event that $|z_i + z_j| - |z_i - z_j| \le 2\kappa$, and analyze the cases in detail. First consider the case $z_i < -\kappa$:

$$
\begin{aligned}
\mathbf{Pr}\left[E, z_i + z_j \ge 0, z_i - z_j \ge 0 \mid z_i < -\kappa\right] &= \mathbf{Pr}\left[z_j \le z_i, z_j \ge -z_i \mid z_i < -\kappa\right] = 0,\\
\mathbf{Pr}\left[E, z_i + z_j \ge 0, z_i - z_j < 0 \mid z_i < -\kappa\right] &= \mathbf{Pr}\left[z_j > |z_i|, z_i \le \kappa \mid z_i < -\kappa\right] = \Phi(z_i),\\
\mathbf{Pr}\left[E, z_i + z_j < 0, z_i - z_j \ge 0 \mid z_i < -\kappa\right] &= \mathbf{Pr}\left[z_j < -|z_i|, z_i \ge -\kappa \mid z_i < -\kappa\right] = 0,\\
\mathbf{Pr}\left[E, z_i + z_j < 0, z_i - z_j < 0 \mid z_i < -\kappa\right] &= \mathbf{Pr}\left[z_i < z_j < -z_i, z_j > -\kappa \mid z_i < -\kappa\right]\\
&= \Phi(\kappa) - \Phi(z_i).
\end{aligned}
$$

The sum of the four probabilities in the above is $\mathbf{Pr}\left[E \mid z_i < -\kappa\right] = \Phi(\kappa)$. Similarly, we analyze the other case, $z_i > \kappa$:

$$
\begin{aligned}
\mathbf{Pr}\left[E, z_i + z_j \ge 0, z_i - z_j \ge 0 \mid z_i > \kappa\right] &= \mathbf{Pr}\left[-z_i \le z_j \le z_i, z_j \le \kappa \mid z_i > \kappa\right]\\
&= \Phi(\kappa) - \Phi_{\mathrm{c}}(z_i),\\
\mathbf{Pr}\left[E, z_i + z_j \ge 0, z_i - z_j < 0 \mid z_i > \kappa\right] &= \mathbf{Pr}\left[z_j > |z_i|, z_i \le \kappa \mid z_i > \kappa\right] = 0,\\
\mathbf{Pr}\left[E, z_i + z_j < 0, z_i - z_j \ge 0 \mid z_i > \kappa\right] &= \mathbf{Pr}\left[z_j < -|z_i|, z_i \ge -\kappa \mid z_i > \kappa\right] = \Phi_{\mathrm{c}}(z_i),\\
\mathbf{Pr}\left[E, z_i + z_j < 0, z_i - z_j < 0 \mid z_i > \kappa\right] &= \mathbf{Pr}\left[z_j < -z_i, z_j > z_i \mid z_i > \kappa\right] = 0.
\end{aligned}
$$

The sum of these four probabilities is $\mathbf{Pr}\left[E \mid z_i > \kappa\right] = \Phi(\kappa)$. Therefore, we obtain

$$
\mathbf{Pr}\left[|z_i + z_j| - |z_i - z_j| \le 2\kappa \mid |z_i| > \kappa\right] = \Phi(\kappa),
$$

which justifies (4.6).

Next, note that $\mathbf{Pr}\left[|z_i| \le \kappa\right] = \Phi(\kappa) - \Phi_{\mathrm{c}}(\kappa)$ and $\mathbf{Pr}\left[|z_i| > \kappa\right] = 2\Phi_{\mathrm{c}}(\kappa)$, so we have from (4.6) that

$$
\begin{aligned}
\mathbf{Pr}\left[h^*(\mathbf{X}'_{ij}) = 0\right] &\le \Phi(\kappa) - \Phi_{\mathrm{c}}(\kappa) + 2\Phi_{\mathrm{c}}(\kappa)\Phi(\kappa)\\
&= 1 - 2\Phi_{\mathrm{c}}(\kappa) + 2\Phi_{\mathrm{c}}(\kappa)\Phi(\kappa) = 1 - 2\Phi_{\mathrm{c}}(\kappa)^2.
\end{aligned}
$$

Thus, $\mathbf{X}'_{ij}$ is misclassified with probability at least $2\Phi_{\mathrm{c}}(\kappa)^2$.

We will now construct sets of pairs with mutually independent elements, such that the union of those sets covers all inter-class edges. This will enable us to use a concentration argument that computes the fraction of the misclassified inter-class edges. Since the graph operations are permutation invariant, let us assume for simplicity that $C_0 = \{1, \ldots, \frac{n}{2}\}$ and $C_1 = \{\frac{n}{2} + 1, \ldots, n\}$ for an even number of nodes n. Also, define the function

$$m(i,l) = \begin{cases} i+l, & i+l \leq \frac{n}{2}, \\ i+l-\frac{n}{2}, & i+l > \frac{n}{2}. \end{cases}.$$

We now construct the following sequence of sets for all $l \in \{0, \ldots, \frac{n}{2} - 1\}$:

$$S_l = \{(X_{m(i,l)}, X_{i+\frac{n}{2}}) \text{ for all } i \in C_0 \text{ such that } (m(i,l), i+n/2) \in E\}.$$

Fix $l \in \{0, \ldots, \frac{n}{2} - 1\}$ and observe that the pairs in the set S_l are mutually independent. Define a Bernoulli random variable, β_i, to be the indicator that $(X_{m(i,l)}, X_{i+\frac{n}{2}})$ is misclassified. We have that $\mathbb{E}[\beta_i] \geq 2\Phi_{\mathrm{c}}(\kappa)^2$. Note that the fraction of pairs in the set S_l that are misclassified is $\frac{1}{|S_l|}\sum_{i:(X_{m(i,l)}, X_{i+n/2}) \in S_l} \beta_i$, which is a sum of independent Bernoulli random variables. Hence, by the additive Chernoff bound, we obtain

$$\mathbf{Pr}\left[\sum_{i \in C_0 \cap N_{m(i,l)}} \beta_i \geq 2|S_l|\Phi_{\mathrm{c}}(\kappa)^2 - |S_l|t\right] \geq 1 - \exp(-2|S_l|t^2).$$

Since $p, q = \Omega(\frac{\log^2 n}{n})$, we have by the Chernoff bound and a union bound that with probability at least $1 - 1/\mathrm{poly}(n)$, $|S_l| = nq(1 \pm o(1))$ for all l. We now choose $t = \sqrt{\frac{C \log n}{|S_l|}} = o(1)$ to obtain that on the event where $|S_l| = nq(1 \pm o(1))$, we have the following for any large $C > 1$:

$$\mathbf{Pr}\left[\frac{1}{|S_l|}\sum_{i \in C_0 \cap N_{m(i,l)}} \beta_i \geq 2\Phi_{\mathrm{c}}(\kappa)^2 - o(1)\right] \geq 1 - n^{-C}.$$

Following a union bound over all $l \in \{0, \ldots, \frac{n}{2} - 1\}$, we conclude that for any $c > 0$,

$$\mathbf{Pr}\left[\frac{1}{|S_l|}\sum_{i \in C_0 \cap N_{m(i,l)}} \beta_i \geq 2\Phi_{\mathrm{c}}(\kappa)^2 - o(1), \;\; \forall l \in \left\{0, \ldots, \frac{n}{2} - 1\right\}\right] \geq 1 - O(n^{-c}).$$

Thus, out of all the pairs $\mathbf{X}'_{ij}$ with $j \nsim i$, with probability at least $1 - o(1)$, we have that at least a fraction $2\Phi_{\mathrm{c}}(\kappa)^2$ of the pairs are misclassified by the attention mechanism. This concludes part 1 of the theorem.

For part 2, note that by the additive Chernoff bound we have for any $t \in (0,1)$,

$$\mathbf{Pr}\left[\sum_{i \in C_0 \cap N_{m(i,l)}} \beta_i \geq 2|S_l|\Phi_{\mathrm{c}}(\kappa)^2 - |S_l|t\right] \geq 1 - \exp(-2|S_l|t^2).$$

Since $|S_l| = \frac{nq}{2}(1 \pm o(1))$ with probability at least $1/\mathrm{poly}(n)$, we choose $t = 2\sqrt{\frac{K\Phi_{\mathrm{c}}(\kappa)^2 \log^2 n}{nq}}$ to obtain

$$\mathbf{Pr}\left[\sum_{i \in C_0 \cap N_{m(i,l)}} \beta_i \geq nq\Phi_{\mathrm{c}}(\kappa)^2(1 \pm o(1)) - \sqrt{Knq\Phi_{\mathrm{c}}(\kappa)^2 \log^2 n}\right] \geq 1 - O(n^{-8K\Phi_{\mathrm{c}}(\kappa)^2 \log n}).$$

Now note that if $q > \frac{K \log^2 n}{n\Phi_{\mathrm{c}}(\kappa)^2}$ then we have $nq\Phi_{\mathrm{c}}(\kappa)^2 > K \log^2 n$, which implies that

$$nq\Phi_{\mathrm{c}}(\kappa)^2 - \sqrt{Knq\Phi_{\mathrm{c}}(\kappa)^2 \log^2 n} > 0.$$

Hence, in this regime of q,

$$\mathbf{Pr}\left[\sum_{i \in C_0 \cap N_{m(i,l)}} \beta_i > 0\right] \geq 1 - O(n^{-8K\Phi_{\mathrm{c}}(\kappa)^2 \log n}).$$

This completes the proof. □

4.4 Attention Coefficients

As a motivating example of how the attention mechanism would fail and how exactly the attention coefficients would behave in this regime, we focus on one of the most popular attention architecture [VCC+18a], where α is a single-layer neural network parameterized by $(\mathbf{w}, \boldsymbol{a}, b) \in \mathbb{R}^d \times \mathbb{R}^2 \times \mathbb{R}$ with LeakyRelu activation function. Namely, the attention coefficients are defined by

$$\tau_{ij} \stackrel{\mathrm{def}}{=} \frac{\exp\left(\mathrm{LeakyRelu}\left(\boldsymbol{a}^\top \begin{bmatrix} \mathbf{w}^\top \mathbf{X}_i \\ \mathbf{w}^\top \mathbf{X}_j \end{bmatrix} + b\right)\right)}{\sum_{\ell \in \eta_1(i)} \exp\left(\mathrm{LeakyRelu}\left(\boldsymbol{a}^\top \begin{bmatrix} \mathbf{w}^\top \mathbf{X}_i \\ \mathbf{w}^\top \mathbf{X}_\ell \end{bmatrix} + b\right)\right)}. \tag{4.7}$$

We show that, as a consequence of the inability of the attention mechanism to distinguish intra-class and inter-class edges, with overwhelming probability most of the attention coefficients τ_{ij} given by (4.7) are going to be $\Theta(1/|\eta_1(i)|)$. In particular, Theorem 6 says that for the vast majority of nodes in the graph, the attention coefficients on most edges are uniform irrespective of whether the edge is inter-class or intra-class. As a result, this means that the attention mechanism is unable to assign higher weights to important edges and lower weights to unimportant edges.

Theorem 6. *Assume that $\|\boldsymbol{\mu}\| \leq K\sigma$ and $\sigma \leq K'$ for some absolute constants K and K', and the parameters $(\mathbf{w}, \boldsymbol{a}, b) \in \mathbb{R}^d \times \mathbb{R}^2 \times \mathbb{R}$ are bounded. Then, with probability at least $1 - o(1)$ over the data $(\mathbf{A}, \mathbf{X}) \sim \mathrm{CSBM}(n, p, q, \boldsymbol{\mu}, \sigma)$, there exists a subset $\mathcal{A} \subseteq [n]$ with cardinality at least $n(1 - o(1))$ such that for all $i \in \mathcal{A}$ the following hold:*

1. *There is a subset $J_{i,0} \subseteq \eta_1(i) \cap C_0$ with cardinality at least $\frac{9}{10}|\eta_1(i) \cap C_0|$, such that $\tau_{ij} = \Theta(1/|\eta_1(i)|)$ for all $j \in J_{i,0}$.*
2. *There is a subset $J_{i,1} \subseteq \eta_1(i) \cap C_1$ with cardinality at least $\frac{9}{10}|\eta_1(i) \cap C_1|$, such that $\tau_{ij} = \Theta(1/|\eta_1(i)|)$ for all $j \in J_{i,1}$.*

Theorem 6 is proved by carefully computing the numerator and the denominator in (4.7). In this regime, $\|\boldsymbol{\mu}\|$ is not much larger than σ, that is, the signal does not dominate noise, so the numerator in (4.7) is not indicative of the class memberships of nodes i, j but rather acts like Gaussian noise. On the other hand, denote the denominator in (4.7) by δ_i and observe that it is the same for all $\tau_{i\ell}$ where $\ell \in \eta_1(i)$. Using concentration arguments about $\{\mathbf{w}^\top \mathbf{X}_\ell\}_{\ell \in [n]}$ yields $\tau_{ij} = \Theta(1/\delta_i)$ and $\delta_i = \Theta(|\eta_1(i)|)$ finishes up the proof.

4.5 Proof of Theorem 6

The proof requires the following assumption on graph sparsity.

Assumption 1. *$p, q = \Omega(\log^2 n/n)$.*

Let us now define the following high-probability events which will be used in the proof. Each of these events holds with probability at least $1 - o(1)$, which follows from straightforward applications of Chernoff bound and union bound, e.g., see [BFJ21a].

Definition 4.5.1. Define the following events over $\mathbf{A}, \mathbf{X}$ and $\{\epsilon_i\}_{i \in [n]}$:

- $\boldsymbol{\mathcal{E}}_1$ is the event that $|C_0| = \frac{n}{2} \pm O(\sqrt{n \log n})$ and $|C_1| = \frac{n}{2} \pm O(\sqrt{n \log n})$.
- $\boldsymbol{\mathcal{E}}_2$ is the event that for each $i \in [n]$, $\mathbf{D}_{ii} = \frac{n(p+q)}{2}\left(1 \pm \frac{10}{\sqrt{\log n}}\right)$.
- $\boldsymbol{\mathcal{E}}_3$ is the event that for each $i \in [n]$, $|C_0 \cap N_i| = \mathbf{D}_{ii} \cdot \frac{(1-\epsilon_i)p+\epsilon_i q}{p+q}\left(1 \pm \frac{10}{\sqrt{\log n}}\right)$ and $|C_1 \cap N_i| = \mathbf{D}_{ii} \cdot \frac{(1-\epsilon_i)q+\epsilon_i p}{p+q}\left(1 \pm \frac{10}{\sqrt{\log n}}\right)$.
- $\boldsymbol{\mathcal{E}}_4$ is the event that for each $i \in [n]$, $\left|\tilde{\mathbf{w}}^\top \mathbf{X}_i - \mathbb{E}\left[\tilde{\mathbf{w}}^\top \mathbf{X}_i\right]\right| \leq 10\sigma\sqrt{\log n}$.
- $\boldsymbol{\mathcal{E}}^*$ is the intersection of the above 4 events.

For $i \in [n]$ let us write $\mathbf{X}_i = (2\epsilon_i - 1)\boldsymbol{\mu} + \sigma \boldsymbol{g}_i$ where $\boldsymbol{g}_i \sim \mathcal{N}(0, \mathbf{I})$, $\epsilon_i = 0$ if $i \in C_0$ and $\epsilon_i = 1$ if $i \in C_1$. Moreover, since the parameters $(\mathbf{w}, \boldsymbol{a}, b) \in \mathbb{R}^d \times \mathbb{R}^2 \times \mathbb{R}$ are bounded, we can write $\mathbf{w} = R\hat{\mathbf{w}}$ and $\boldsymbol{a} = R'\hat{\boldsymbol{a}}$ such that $\|\hat{\mathbf{w}}\| = 1$ and $\|\hat{\boldsymbol{a}}\| = 1$ and R, R' are some constants. We define the following sets which will become useful later in our computation of τ_{ij}'s. Define

$$\mathcal{A} \stackrel{\text{def}}{=} \left\{ i \in [n] \;\middle|\; \begin{array}{l} |\hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i| \leq 10\sqrt{\log(n(p+q))}, \text{ and} \\ |\hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_j| \leq 10\sqrt{\log(n(p+q))}, \ \forall j \in \eta_1(i) \end{array} \right\}.$$

For $i \in [n]$ define

$$\begin{aligned}
J_{i,0} &\stackrel{\text{def}}{=} \left\{ j \in \eta_1(i) \cap C_0 \mid |\hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_j| \leq \sqrt{10} \right\}, \\
J_{i,1} &\stackrel{\text{def}}{=} \left\{ j \in \eta_1(i) \cap C_1 \mid |\hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_j| \leq \sqrt{10} \right\}, \\
B_{i,0}^t &\stackrel{\text{def}}{=} \left\{ j \in \eta_1(i) \cap C_0 \mid 2^{t-1} \leq \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_j \leq 2^t \right\}, \ t = 1, 2, \ldots, T, \\
B_{i,1}^t &\stackrel{\text{def}}{=} \left\{ j \in \eta_1(i) \cap C_1 \mid 2^{t-1} \leq \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_j \leq 2^t \right\}, \ t = 1, 2, \ldots, T,
\end{aligned}$$

where $T \stackrel{\text{def}}{=} \left\lceil \log_2\left(10\sqrt{\log(n(p+q))}\right) \right\rceil$.

We start with a few claims about the sizes of these sets.

Claim 4.5.1. *With probability at least* $1 - o(1)$, *we have that* $|\mathcal{A}| \geq n(1 - o(1))$.

Proof. Because $|\hat{\boldsymbol{a}}_2| \leq 1$ we know that $\mathcal{A}$ is a superset of $\mathcal{A}'$ where

$$\mathcal{A}' \stackrel{\text{def}}{=} \left\{ i \in [n] \;\middle|\; \begin{array}{l} |\hat{\mathbf{w}}^\top \boldsymbol{g}_i| \leq 10\sqrt{\log(n(p+q))}, \text{ and} \\ |\hat{\mathbf{w}}^\top \boldsymbol{g}_j| \leq 10\sqrt{\log(n(p+q))}, \ \forall j \in \eta_1(i) \end{array} \right\}.$$

We give a lower bound for $|\mathcal{A}'|$ and prove the result. First of all, note that if $p + q \geq \Omega(1/\log^2 n)$, then $\log(n(p+q)) = \log n(1 - o(1))$ and we easily get that with probability at least $1 - o(1)$, $|\hat{\mathbf{w}}^\top \boldsymbol{g}_i| \leq 10\sqrt{\log(n(p+q))}$ for all $i \in [n]$, and thus $|\mathcal{A}| = |\mathcal{A}'| = n$. Therefore let us assume without loss of generality that $p + q \leq O(1/\log^2 n)$. Consider the following sum of indicator random variables

$$S \stackrel{\text{def}}{=} \sum_{i\in[n]} \mathbf{1}_{\left\{|\hat{\mathbf{w}}^\top \boldsymbol{g}_i| \geq 10\sqrt{\log(n(p+q))}\right\}}.$$

By the multiplicative Chernoff bound, for any $\delta > 0$ we have

$$\mathbf{Pr}\left[S \geq nb(1+\delta)\right] \leq \exp\left(-\frac{\delta^2}{2+\delta} nb\right)$$

where $b \stackrel{\text{def}}{=} \mathbf{Pr}\left[(|\,|\hat{\mathbf{w}}^\top \boldsymbol{g}_i| \geq 10\sqrt{\log(n(p+q))})\right)$. Moreover, by the standard upper bound on the Gaussian tail probability (Proposition 2.1.2, [Ver18]) we know that $b < e^{-50\log(n(p+q))}$. Let us set

$$\delta \stackrel{\text{def}}{=} \frac{1}{bn(p+q)\log n}.$$

Then by the upper bound on b and the assumption that $p, q = \Omega(\log^2 n/n)$ we know that

$$\delta \geq \frac{(n(p+q))^{49}}{\log n} \geq \Omega(\log^{97} n) = \omega(1).$$

It follows that

$$\frac{\delta^2}{2+\delta} nb \geq \Omega(\delta nb) = \Omega\left(\frac{1}{(p+q)\log n}\right) \geq \Omega(\log n).$$

Therefore, with probability at least $1 - o(1)$ we have that

$$S \leq nb(1+\delta) \leq \frac{n}{(n(p+q))^{50}} + \frac{n}{n(p+q)\log n} = O\left(\frac{n}{n(p+q)\log n}\right).$$

Apply the concentration result of node degrees, this means that with probability at least $1 - o(1)$, for $\delta = 10\sqrt{\log(n(p+q))}$,

$$\begin{aligned}
&\left|\left\{i \in [n] \;\middle|\; |\hat{\mathbf{w}}^\top \boldsymbol{g}_i| \geq \delta \text{ or } \exists j \in \eta_1(i) \text{ such that } |\hat{\mathbf{w}}^\top \boldsymbol{g}_j| \geq \delta\right\}\right| \\
\leq\; & S \cdot \frac{n}{2}(p+q)(1 \pm o(1)) = O\left(\frac{n}{n(p+q)\log n}\right) \cdot \frac{n}{2}(p+q)(1 \pm o(1)) = O\left(\frac{n}{\log n}\right).
\end{aligned}$$

Therefore we have

$$|\mathcal{A}'| \geq n - O(n/\log n) = n(1 - o(1)).$$

□

Claim 4.5.2. *With probability at least $1 - o(1)$, we have that for all $i \in [n]$,*

$$|J_{i,0}| \geq \frac{9}{10}|\eta_1(i) \cap C_0| \quad \textit{and} \quad |J_{i,1}| \geq \frac{9}{10}|\eta_1(i) \cap C_1|.$$

Proof. We prove the result for $J_{i,0}$, the result for $J_{i,1}$ follows analogously. First, fix $i \in [n]$. For each $j \in |\eta_1(i) \cap C_0|$ we have that

$$\mathbf{Pr}[|\hat{\boldsymbol{a}}_2 \mathbf{w}^\top \boldsymbol{g}_j| \geq \sqrt{10}] \leq \mathbf{Pr}[|\mathbf{w}^\top \boldsymbol{g}_j| \geq \sqrt{10}] \leq e^{-50}.$$

Denote $J_{i,0}^c \overset{\text{def}}{=} (\eta_1(i) \cap C_0) \setminus J_{i,0}$. We have that

$$\mathbb{E}[|J_{i,0}^c|] = \mathbb{E}\left[\sum_{j \in \eta_1(i) \cap C_0} \mathbf{1}_{\left\{|\hat{\boldsymbol{a}}_2 \mathbf{w}^\top \boldsymbol{g}_j| \geq \sqrt{10}\right\}}\right] \leq e^{-50}|\eta_1(i) \cap C_0|,$$

Apply Chernoff's inequality (Theorem 2.3.4 in [Ver18]) we have

$$\begin{aligned}
\mathbf{Pr}\left[|J_{i,0}^c| \geq \frac{1}{10}|\eta_1(i) \cap C_0|\right] &\leq e^{-\mathbb{E}[|J_{i,0}^c|]}\left(\frac{e\mathbb{E}[|J_{i,0}^c|]}{|\eta_1(i) \cap C_0|/10}\right)^{|\eta_1(i) \cap C_0|/10} \\
&\leq \left(\frac{ee^{-50}|\eta_1(i) \cap C_0|}{|\eta_1(i) \cap C_0|/10}\right)^{|\eta_1(i) \cap C_0|/10} \\
&= \exp\left(-\left(\frac{1}{2} - \frac{\log 10}{10} - \frac{1}{10}\right)|\eta_1(i) \cap C_0|\right) \\
&\leq \exp\left(-\frac{4}{25}|\eta_1(i) \cap C_0|\right).
\end{aligned}$$

Apply the union bound we get

$$\begin{aligned}
&\mathbf{Pr}\left[|J_{i,0}| \geq \frac{9}{10}|C_0 \cap \eta_1(i)|, \forall i \in [n]\right] \geq 1 - \sum_{i \in [n]} \exp\left(-\frac{4}{25}|\eta_1(i) \cap C_0|\right) \\
&\geq \mathbf{Pr}\left[(] \boldsymbol{\mathcal{E}}_3\right) \cdot \left(1 - \sum_{i \in [n]} \exp\left(-\frac{4}{25}\frac{n\min(p,q)(1-o(1))}{2}\right)\right) \\
&= (1 - o(1)) \cdot \left(1 - n\exp\left(-\frac{2n\min(p,q)(1-o(1))}{25}\right)\right) \\
&= 1 - o(1).
\end{aligned}$$

The second inequality follows because $|\eta_1(i) \cap C_0| \geq \frac{n}{2}\min(p,q)(1-o(1))$ under the event $\boldsymbol{\mathcal{E}}_3$ (cf. Definition 4.5.1) for all $i \in [n]$. The last equality is due to our assumption that $p, q = \Omega(\frac{\log^2 n}{n})$. ☐

Claim 4.5.3. *With probability at least $1 - o(1)$, we have that for all $i \in [n]$ and for all $t \in [T]$,*

$$|B_{i,0}^t| \leq \mathbb{E}[|B_{i,0}^t|] + \sqrt{T}|\eta_1(i) \cap C_0|^{\frac{4}{5}} \quad \textit{and} \quad |B_{i,1}^t| \leq \mathbb{E}[|B_{i,1}^t|] + \sqrt{T}|\eta_1(i) \cap C_1|^{\frac{4}{5}}.$$

Proof. We prove the result for $B_{i,0}^t$, and the result for $B_{i,1}^t$ follows analogously. First fix $i \in [n]$ and $t \in [T]$. By the additive Chernoff inequality, we have

$$\mathbf{Pr}\left[|B_{i,0}^t| \geq \mathbb{E}[|B_{i,0}^t|] + |\eta_1(i) \cap C_0| \cdot \sqrt{T}|\eta_1(i) \cap C_0|^{-\frac{1}{5}}\right] \leq e^{-2T|\eta_1(i)\cap C_0|^{3/5}}.$$

Taking a union bound over all $i \in [n]$ and $t \in [T]$ we get

$$\begin{aligned} &\mathbf{Pr}\left[\bigcup_{i\in[n]}\bigcup_{t\in[T]}\left\{|B_{i,0}^t| \geq \mathbb{E}[|B_{i,0}^t|] + \sqrt{T}|\eta_1(i) \cap C_0|^{\frac{4}{5}}\right\}\right] \\ &\leq \; nT\exp\left(-2T\left(\frac{n}{2}\min(p,q)(1-o(1))\right)^{3/5}\right) + o(1) \;=\; o(1), \end{aligned}$$

where the last equality follows from Assumption 1 that $p, q = \Omega(\frac{\log^2 n}{n})$, and hence

$$nT\exp\left(-2T\left(\frac{n}{2}\min(p,q)(1-o(1))\right)^{3/5}\right) = nT\exp\left(-\omega\left(\sqrt{2T}\log n\right)\right) = O\left(n^{-c}\right)$$

for some absolute constant $c > 0$. Moreover, we have used degree concentration, which introduced the additional additive $o(1)$ term in the probability upper bound. Therefore we have

$$\mathbf{Pr}\left[|B_{i,0}^t| \leq \mathbb{E}[|B_{i,0}^t|] + \sqrt{T}|\eta_1(i) \cap C_0|^{\frac{4}{5}}, \forall i \in [n] \; \forall t \in [T]\right] \geq 1 - o(1).$$

☐

We start by defining an event $\boldsymbol{\mathcal{E}}^{\#}$ which is the intersection of the following events over the randomness of $\mathbf{A}$ and $\{\epsilon_i\}_i$ and $\mathbf{X}_i = (2\epsilon_i - 1)\boldsymbol{\mu} + \sigma \boldsymbol{g}_i$,

- $\boldsymbol{\mathcal{E}}_1'$ is the event that for each $i \in [n]$, $|C_0 \cap \eta_1(i)| = \frac{n}{2}((1-\epsilon_i)p + \epsilon_i q)(1 \pm o(1))$ and $|C_1 \cap \eta_1(i)| = \frac{n}{2}((1-\epsilon_i)q + \epsilon_i p)(1 \pm o(1))$.

- $\boldsymbol{\mathcal{E}}'_2$ is the event that $|\mathcal{A}| \geq n - o(\sqrt{n})$.
- $\boldsymbol{\mathcal{E}}'_3$ is the event that $|J_{i,0}| \geq \frac{9}{10}|\eta_1(i) \cap C_0|$ and $|J_{i,1}| \geq \frac{9}{10}|\eta_1(i) \cap C_1|$ for all $i \in [n]$.
- $\boldsymbol{\mathcal{E}}'_4$ is the event that $|B^t_{i,0}| \leq \mathbb{E}[|B^t_{i,0}|] + \sqrt{T}|\eta_1(i) \cap C_0|^{\frac{4}{5}}$ and $|B^t_{i,1}| \leq \mathbb{E}[|B^t_{i,1}|] + \sqrt{T}|\eta_1(i) \cap C_1|^{\frac{4}{5}}$ for all $i \in [n]$ and for all $t \in [T]$.

By Claims 4.5.1 to 4.5.3, we get that with probability at least $1 - o(1)$, the event $\boldsymbol{\mathcal{E}}^{\#} \stackrel{\text{def}}{=} \bigcap_{i=1}^{4} \boldsymbol{\mathcal{E}}'_i$ holds. We will show that under event $\boldsymbol{\mathcal{E}}^{\#}$, for all $i \in \mathcal{A}$, for all $j \in J_{i,c}$ where $c \in \{0, 1\}$, we have $\tau_{ij} = \Theta(1/|\eta_1(i)|)$. This will prove Theorem 6.

Fix $i \in \mathcal{A}$ and some $j \in J_{i,0}$. Let us consider

$$\begin{aligned}
\tau_{ij} &= \frac{\exp\left(\mathrm{LeakyRelu}(\boldsymbol{a}_1 \mathbf{w}^\top \mathbf{X}_i + \boldsymbol{a}_2 \mathbf{w}^\top \mathbf{X}_j + b)\right)}{\sum_{k \in \eta_1(i)} \exp\left(\mathrm{LeakyRelu}(\boldsymbol{a}_1 \mathbf{w}^\top \mathbf{X}_i + \boldsymbol{a}_2 \mathbf{w}^\top \mathbf{X}_k + b)\right)} \\
&= \frac{\exp\left(\sigma R R' \; \mathrm{LeakyRelu}(\kappa_{ij} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_j + b')\right)}{\sum_{k \in \eta_1(i)} \exp\left(\sigma R R' \; \mathrm{LeakyRelu}(\kappa_{ik} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k + b')\right)} \\
&= \frac{1}{\sum_{k \in \eta_1(i)} \exp(\Delta_{ik} - \Delta_{ij})}
\end{aligned}$$

where for $l \in \eta_1(i)$, we denote

$$\begin{aligned}
\kappa_{il} &\stackrel{\text{def}}{=} (2\epsilon_i - 1)\hat{\mathbf{w}}^\top \boldsymbol{\mu}/\sigma + (2\epsilon_l - 1)\hat{\mathbf{w}}^\top \boldsymbol{\mu}/\sigma, \\
\Delta_{il} &\stackrel{\text{def}}{=} \sigma R R' \; \mathrm{LeakyRelu}(\kappa_{il} + \hat{\boldsymbol{a}}_1 \mathbf{w}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \mathbf{w}^\top \boldsymbol{g}_l + b'),
\end{aligned}$$

and $b = \sigma R R' b'$. We will show that

$$\sum_{k \in \eta_1(i)} \exp(\Delta_{ik} - \Delta_{ij}) = \Theta(|\eta_1(i)|)$$

and hence conclude that $\tau_{ij} = \Theta(1/|\eta_1(i)|)$. First of all, note that since $\|\boldsymbol{\mu}\| \leq K\sigma$ for some absolute constant K, we know that

$$|\kappa_{il}| \leq \sqrt{2}K = O(1).$$

Let us assume that $\hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i \geq 0$ and consider the following two cases regarding the magnitude of $\hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i$.

Case 1. If $\kappa_{ij} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_j + b' < 0$, then

$$\begin{aligned}
\Delta_{ik} - \Delta_{ij} &= \sigma R R' \Big(\mathrm{LeakyRelu}(\kappa_{ik} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k + b') \\
&\qquad - \mathrm{LeakyRelu}(\kappa_{ij} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_j + b') \Big) \\
&= \sigma R R' \Big(\mathrm{LeakyRelu}(\hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k \pm O(1)) \\
&\qquad - \beta(\kappa_{ij} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_j + b') \Big) \\
&= \sigma R R' \left(\mathrm{LeakyRelu}(\hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k \pm O(1)) \pm O(1) \right) \\
&= \sigma R R' \left(\Theta(\hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k) \pm O(1) \right),
\end{aligned}$$

where β is the slope of $\mathrm{LeakyRelu}(x)$ for $x < 0$. Here, the second equality follows from $|\kappa_{ik} + b'| \le \sqrt{2}K + |b'| = O(1)$ and $\kappa_{ij} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_j + b' < 0$. The third equality follows from

- We have $j \in J_{i,0}$ and hence $|\hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_j| = O(1)$;
- We have $\kappa_{ij} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_j + b' < 0$, so $\hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i < |\kappa_{ij}| + |\hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_j| + |b'| = O(1)$, moreover, because $\hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i \ge 0$, we get that $|\hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i| = O(1)$;
- We have $|\kappa_{ij} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_j + b'| \le |\hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i| + |\hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_j| + |\kappa_{ij} + b'| = O(1) + O(1) + O(1) = O(1)$.

Case 2. If $\kappa_{ij} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_j + b' \ge 0$, then

$$\begin{aligned}
\Delta_{ik} - \Delta_{ij} &= \sigma R R' \Big(\mathrm{LeakyRelu}(\kappa_{ik} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k + b') \\
&\qquad - \mathrm{LeakyRelu}(\kappa_{ij} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_j + b') \Big) \\
&= \sigma R R' \Big(\mathrm{LeakyRelu}(\kappa_{ik} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k + b') \\
&\qquad - \kappa_{ij} - \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i - \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_j - b' \Big) \\
&= \sigma R R' \left(\mathrm{LeakyRelu}(\kappa_{ik} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k + b') - \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i \pm O(1) \right) \\
&\begin{cases} = \sigma R R' \left(\Theta(\hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k) \pm O(1) \right), & \text{if } k \in J_{i,0} \cup J_{i,1} \\ \le \sigma R R' \left(O(\hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k) \pm O(1) \right), & \text{otherwise.} \end{cases}
\end{aligned}$$

To see the last (in)equality in the above, consider the following cases:

1. If $k \in J_{i,0} \cup J_{i,1}$, then there are two cases depending on the sign of $\kappa_{ik} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k + b'$.

    - If $\kappa_{ik} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k + b' \geq 0$, then we have that

    $$\begin{aligned} & \text{LeakyRelu}(\kappa_{ik} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k + b') - \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i \pm O(1) \\ = \; & \kappa_{ik} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k + b' - \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i \pm O(1) \\ = \; & \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k + \kappa_{ik} + b' \pm O(1) \\ = \; & \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k \pm O(1). \end{aligned}$$

    - If $\kappa_{ik} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k + b' < 0$, then because $\hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i \geq 0$ and $|\kappa_{ik} + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k + b'| \leq |\kappa_{ik}| + |\hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k| + |b'| = O(1)$, we know that $\hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i < |\kappa_{ik}| + |\hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k| + |b'| = O(1)$ and $|\kappa_{ik} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k + b'| = O(1)$. Therefore it follows that

    $$\begin{aligned} & \text{LeakyRelu}(\kappa_{ik} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k + b') - \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i \pm O(1) \\ = \; & \text{LeakyRelu}(\pm O(1)) - O(1) \pm O(1) \\ = \; & \pm O(1) \\ = \; & \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k \pm O(1) \end{aligned}$$

    where the last equality is due to the fact that $k \in J_{i,0} \cup J_{i,1}$ so $|\hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k| = O(1)$.

2. If $k \notin J_{i,0} \cup J_{i,1}$, then there are two cases depending on the sign of $\kappa_{ik} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k + b'$.

    - If $\kappa_{ik} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k + b' \geq 0$, then we have that

    $$\begin{aligned} & \text{LeakyRelu}(\kappa_{ik} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k + b') - \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i \pm O(1) \\ = \; & \kappa_{ik} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k + b' - \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i \pm O(1) \\ = \; & \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k + \kappa_{ik} + b' \pm O(1) \\ = \; & \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k \pm O(1). \end{aligned}$$

    - If $\kappa_{ik} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k + b' < 0$, then we have that,

    $$\begin{aligned} & \text{LeakyRelu}(\kappa_{ik} + \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k + b') - \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i \pm O(1) \\ = \; & \beta\kappa_{ik} + \beta\hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i + \beta\hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k + \beta b' - \hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i \pm O(1) \\ = \; & \beta\hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k - (1-\beta)\hat{\boldsymbol{a}}_1 \hat{\mathbf{w}}^\top \boldsymbol{g}_i \pm O(1) \\ \leq \; & \beta\hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k \pm O(1), \end{aligned}$$

    where β is the slope of LeakyRelu$(\cdot)$.

Combining the two cases regarding the magnitude of $\hat{\boldsymbol{a}}_1\hat{\mathbf{w}}^\top\boldsymbol{g}_i$, so far we have showed that, for any i such that $\hat{\boldsymbol{a}}_1\hat{\mathbf{w}}^\top\boldsymbol{g}_i \geq 0$, for all $j \in J_{i,0}$, we have

$$\Delta_{ik} - \Delta_{ij} = \begin{cases} \Theta(\hat{\boldsymbol{a}}_2\hat{\mathbf{w}}^\top\boldsymbol{g}_k) \pm O(1), & \text{if } k \in J_{i,0} \cup J_{i,1} \\ O(\hat{\boldsymbol{a}}_2\hat{\mathbf{w}}^\top\boldsymbol{g}_k) \pm O(1), & \text{otherwise.} \end{cases} \tag{4.8}$$

By following a similar argument, one can show that Equation 4.8 holds for any i such that $\hat{\boldsymbol{a}}_1\hat{\mathbf{w}}^\top\boldsymbol{g}_i < 0$.

Let us now compute

$$\sum_{k\in\eta_1(i)} \exp(\Delta_{ik} - \Delta_{ij}) = \sum_{k\in\eta_1(i)\cap C_0} \exp(\Delta_{ik} - \Delta_{ij}) + \sum_{k\in\eta_1(i)\cap C_1} \exp(\Delta_{ik} - \Delta_{ij})$$

for some $j \in J_{i,0}$. Let us focus on $\sum_{k\in\eta_1(i)\cap C_0} \exp(\Delta_{ik} - \Delta_{ij})$ first. We will show that $\Omega(|\eta_1(i) \cap C_0|) \leq \sum_{k\in\eta_1(i)\cap C_0} \exp(\Delta_{ik} - \Delta_{ij}) \leq O(|\eta_1(i)|)$.

First of all, we have that

$$\begin{aligned} \sum_{k\in\eta_1(i)\cap C_0} \exp(\Delta_{ik} - \Delta_{ij}) &\geq \sum_{k\in J_{i,0}} \exp(\Delta_{ik} - \Delta_{ij}) = \sum_{k\in J_{i,0}} \exp\big(\Theta(\hat{\boldsymbol{a}}_2\hat{\mathbf{w}}^\top\boldsymbol{g}_k) \pm O(1)\big) \\ &\geq \sum_{k\in J_{i,0}} e^{c_1} = |J_{i,0}|e^{c_1} = \Omega(|\eta_1(i) \cap C_0|), \end{aligned} \tag{4.9}$$

where c_1 is an absolute constant (possibly negative). On the other hand, consider the following partition of $\eta_1(i) \cap C_0$:

$$\begin{aligned} P_1 &\stackrel{\text{def}}{=} \{k \in \eta_1(i) \cap C_0 \mid \hat{\boldsymbol{a}}_2\hat{\mathbf{w}}^\top\boldsymbol{g}_k \leq 1\}, \\ P_2 &\stackrel{\text{def}}{=} \{k \in \eta_1(i) \cap C_0 \mid \hat{\boldsymbol{a}}_2\hat{\mathbf{w}}^\top\boldsymbol{g}_k \geq 1\}. \end{aligned}$$

It is easy to see that

$$\sum_{k\in P_1} \exp(\Delta_{ik} - \Delta_{ij}) \leq \sum_{k\in P_1} \exp\big(O(\hat{\boldsymbol{a}}_2\hat{\mathbf{w}}^\top\boldsymbol{g}_k) \pm O(1)\big) \leq \sum_{k\in P_1} e^{c_2} = |P_1|e^{c_2} = O(|\eta_1(i) \cap C_0|), \tag{4.10}$$

where c_2 is an absolute constant. Moreover, because $i \in \mathcal{A}$ we have that $P_2 \subseteq \bigcup_{t\in[T]} B_{i,0}^t$. It follows that

$$\begin{aligned} \sum_{k\in P_2} \exp(\Delta_{ik} - \Delta_{ij}) &= \sum_{t\in[T]} \sum_{k\in B_{i,0}^t} \exp(\Delta_{ik} - \Delta_{ij}) \\ &\leq \sum_{t\in[T]} \sum_{k\in B_{i,0}^t} \exp\big(O(\hat{\boldsymbol{a}}_2\hat{\mathbf{w}}^\top\boldsymbol{g}_k) \pm O(1)\big) \\ &\leq \sum_{t\in[T]} |B_{i,0}^t|e^{c_3 2^t}, \end{aligned} \tag{4.11}$$

where c_3 is an absolute constant. We can upper bound the above quantity as follows. Under the Event $\boldsymbol{\mathcal{E}}^*$, we have that

$$|B_{i,0}^t| \leq m_t + \sqrt{T}|\eta_1(i) \cap C_0|^{\frac{4}{5}}, \text{ for all } t \in [T],$$

where

$$\begin{aligned} m_t \overset{\text{def}}{=} \mathbb{E}[|B_{i,0}^t|] = \sum_{k \in \eta_1(i) \cap C_0} \mathbf{Pr}\left[(\right] 2^{t-1} \leq \hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k \leq 2^t) \leq \sum_{k \in \eta_1(i) \cap C_0} \mathbf{Pr}[\hat{\boldsymbol{a}}_2 \hat{\mathbf{w}}^\top \boldsymbol{g}_k \geq 2^{t-1}] \\ \leq \sum_{k \in \eta_1(i) \cap C_0} \mathbf{Pr}[\hat{\mathbf{w}}^\top \boldsymbol{g}_k \geq 2^{t-1}] \leq |\eta_1(i) \cap C_0| e^{-2^{2t-3}}. \end{aligned}$$

It follows that

$$\begin{aligned} \sum_{t \in [T]} |B_{i,0}^t| e^{c_3 2^t} &\leq \sum_{t \in [T]} \left(|\eta_1(i) \cap C_0| e^{-2^{2t-3}} + \sqrt{T}|\eta_1(i) \cap C_0|^{\frac{4}{5}} \right) e^{c_3 2^t} \\ &\leq |\eta_1(i) \cap C_0| \sum_{t=1}^{\infty} e^{-2^{2t-3}} e^{c_3 2^t} + \sum_{t \in [T]} \sqrt{T}|\eta_1(i) \cap C_0|^{\frac{4}{5}} e^{c_3 2^\top} \qquad (4.12) \\ &\leq c_4 |\eta_1(i) \cap C_0| + o(|\eta_1(i)|) \\ &\leq O(|\eta_1(i)|), \end{aligned}$$

where c_4 is an absolute constant. The third inequality in the above follows from

- The series $\sum_{t=1}^{\infty} e^{-2^{2t-3}} e^{c_3 2^t}$ converges absolutely for any constant c_3;
- The sum $\sum_{t \in [T]} \sqrt{T}|\eta_1(i) \cap C_0|^{\frac{4}{5}} e^{c_3 2^\top} = T^{\frac{3}{2}} |\eta_1(i) \cap C_0|^{\frac{4}{5}} e^{c_3 2^\top} = o(|\eta_1(i)|)$ because

  $$\begin{aligned} \log\left(T^{\frac{3}{2}} e^{c_3 2^\top}\right) &= \frac{3}{2} \log \left\lceil \log_2 \left(10\sqrt{\log(n(p+q))}\right) \right\rceil + c_3 2^{\left\lceil \log_2\left(10\sqrt{\log(n(p+q))}\right)\right\rceil} \\ &\leq \frac{3}{2} \log \left\lceil \log_2 \left(10\sqrt{\log(n(p+q))}\right) \right\rceil + 20 c_3 \sqrt{\log(n(p+q))} \\ &\leq O\left(\frac{1}{c} \log(n(p+q))\right), \end{aligned}$$

  for any $c > 0$. In particular, by picking $c > 5$ we see that $T^{\frac{3}{2}} e^{c_3 2^\top} \leq O((n(p+q))^{\frac{1}{c}}) \leq o(|\eta_1(i)|^{\frac{1}{5}})$, and hence we get $T^{\frac{3}{2}} e^{c_3 2^\top} |\eta_1(i) \cap C_0|^{\frac{4}{5}} \leq |\eta_1(i)|^{\frac{4}{5}} \cdot o(|\eta_1(i)|^{\frac{1}{5}}) = o(|\eta_1(i)|)$.

Combining Equations 4.11 and 4.12 we get

$$\sum_{k \in P_2} \exp(\Delta_{ik} - \Delta_{ij}) \leq O(|\eta_1(i)|), \tag{4.13}$$

and combining Equations 4.10 and 4.13 we get

$$\sum_{k \in \eta_1(i) \cap C_0} \exp(\Delta_{ik} - \Delta_{ij}) = \sum_{k \in P_1} \exp(\Delta_{ik} - \Delta_{ij}) + \sum_{k \in P_1} \exp(\Delta_{ik} - \Delta_{ij}) \leq O(|\eta_1(i)|). \tag{4.14}$$

Now, by Equations 4.9 and 4.14 we get

$$\Omega(|\eta_1(i) \cap C_0|) \leq \sum_{k \in \eta_1(i) \cap C_0} \exp(\Delta_{ik} - \Delta_{ij}) \leq O(|\eta_1(i)|). \tag{4.15}$$

It turns out that repeating the same argument for $\sum_{k \in \eta_1(i) \cap C_1} \exp(\Delta_{ik} - \Delta_{ij})$ yields

$$\Omega(|\eta_1(i) \cap C_1|) \leq \sum_{k \in \eta_1(i) \cap C_1} \exp(\Delta_{ik} - \Delta_{ij}) \leq O(|\eta_1(i)|). \tag{4.16}$$

Finally, Equations 4.15 and 4.16 give us

$$\sum_{k \in \eta_1(i)} \exp(\Delta_{ik} - \Delta_{ij}) = \Theta(|\eta_1(i)|),$$

which readily implies

$$\tau_{ij} = \frac{1}{\sum_{k \in \eta_1(i)} \exp(\Delta_{ik} - \Delta_{ij})} = \Theta(1/|\eta_1(i)|)$$

as required. We have showed that for all $i \in \mathcal{A}$ and all $j \in J_{i,0}$, $\tau_{ij} = \Theta(1/|\eta_1(i)|)$. Repeating the same argument we get that the same result holds for all $i \in \mathcal{A}$ and all $j \in J_{i,1}$, too. Hence, by Claims 4.5.1 and 4.5.2 about the cardinalities of $\mathcal{A}$, $J_{i,0}$ and $J_{i,1}$ we have thus proved Theorem 6.

Chapter 5

Optimality of Message-Passing

In this chapter, I will present a theoretical analysis of the message-passing GNN given in Architecture 2. We begin by defining a natural notion of optimality in our setting and show that among local learning methods on graphs, Architecture 2 is optimal according to this definition on a very general statistical model. We then compute the generalization error and compare the architecture with other well-studied methods like a simple MLP and a GCN. We will view n as large and study the setting where d is fixed (does not grow with n) and work in the extremely sparse setting where $q_{ij} = b_{ij}/n$ for constants $b_{ij} > 1$, so we write $\mathbf{Q} = \mathbf{B}/n$ where $\mathbf{B} = \{b_{ij}\}_{i,j\in[C]}$. Furthermore, the only assumption we need about the distributions $\mathbb{P}_i$ is that $\mathbb{P}_i$ are absolutely continuous with respect to some base measure, in which case their densities exist, denoted by ρ_i. For ease of reading, we encourage the reader to consider the case where $\mathbb{P}_i$ are continuous or discrete, therefore, the base measure is simply the Lebesgue measure on $\mathbb{R}$ or the counting measure on $\mathbb{Z}$ respectively.

5.1 Local Weak Convergence

We briefly recall here the notion of local weak convergence of random rooted graphs. The notion of local weak convergence of random, feature-decorated, rooted graphs is defined analogously. Let us begin first with the case of rooted graphs. A rooted graph $(G, u) = ((E, V), u)$ is a graph G with a distinguished vertex u called the root. We say that two rooted graphs $(G_1, u_1) = ((E_1, V_1), u_1)$, $(G_2, u_2) = ((E_2, V_2), u_2)$ are isomorphic if there is a bijection $\phi : V_1 \to V_2$ such that $\phi(u) = v$ and such that if $(x, y) \in E_1$ then $(\phi(x), \phi(y)) \in E_2$. In this case, we write $(G_1, u_1) \cong (G_2, u_2)$. For a rooted graph (G, u), we denote its isomorphism class as $[(G, u)]$.

Let $\mathcal{G}_*$ denote the set of isomorphism classes of (locally finite) rooted graphs. Here, locally finite means that for constant depth starting from a root node in the graph, the number of nodes in this constant-depth neighbourhood is constant. For a vertex $v \in G$ we let $\eta_k(v)$ denote the collection of neighbours of v of distance at most k in the canonical edge distance metric, and $G[\eta_k(v)]$ denote the subgraph induced on this collection of vertices. We then have the notion of *local convergence* in $\mathcal{G}_*$.

Definition 5.1.1 (Local convergence on $\mathcal{G}_*$)**.** A sequence $[(G_n, u_n)] \in \mathcal{G}_*$ *converges locally* to $[(G, u)] \in \mathcal{G}_*$ if for each $k > 0$, we have that $G_n[\eta_k(u_n)] \cong G[\eta_k(u)]$ for large enough n.

It can be shown [Bor16, Lemma 3.4] that $\mathcal{G}_*$ equipped with the topology of local convergence is a Polish space. We are then in the position to define local weak convergence on $\mathcal{G}_*$. In brief, the topology of local weak convergence of random graphs is the topology of weak convergence of measures on the space of probability measures on $\mathcal{G}_*$, namely $\mathcal{M}_1(G_*)$.

Definition 5.1.2 (Local weak convergence of rooted graphs)**.** A sequence $\{[G_n, u_n]\}$ of random rooted graphs with corresponding laws $\{\mu_n\} \subseteq \mathcal{M}_1(\mathcal{G}_*)$ is said to locally weakly converge to a random rooted graph $[G, u]$ with law $\mu \in \mathcal{M}_1(\mathcal{G}_*)$ if $\mu_n \to \mu$ weakly.

We note here that it is common to talk about the notion of local weak convergence of a sequence of (finite) random graphs G_n. In this case $G_n \to G$ locally weakly if $[(G_n, u_n)] \to [(G, u)]$ locally weakly where $u_n \sim \mathrm{Unif}(V(G_n))$.

5.2 Asymptotic Local Bayes Optimality

For classification tasks, it is natural to use a notion of generalization error in a "per sample" or online sense. Without graphical side information, the natural choice is the Bayes risk. With graphical information, however, there is an important obstruction: the number of samples is equal to the size of the corresponding graph. As such, a naive extension of the Bayes risk does not have this property.

A natural approach would be to consider the Bayes risk for estimators that take in the node, the data set, and the graph, i.e., $\hat{y}_v = \hat{y}(v, (X, G))$. In this case, however, the risk necessarily implicitly depends on the sample size, n, through G. One might try to remove this dependence by taking the infinite sample size limit, but for a class of estimators this general, it is not clear that such a limit is well defined. To circumvent this issue, we restrict attention to node classifiers that are only allowed "local" information around the node. The large graph limit of the generalization error is then naturally interpreted via *local weak convergence*, which is discussed in the following subsection. We refer the reader to [Ram21, Chapter 1] or [Bor16, Section 3] for more detailed expositions.

In this limit, one can then interpret the generalization error as a per-sample error for the randomly rooted graph (G, u) where u is a uniform random vertex in $V(G)$. With these observations, we are led to a natural notion of Bayes optimality, namely *asymptotic local Bayes optimality* which is defined below. First, let us recall the notion of *ℓ-local classifiers*.

Definition 5.2.1 (ℓ-local classifier)**.** Let $G = (\mathbf{A}, \mathbf{X})$ be a feature-decorated graph of n vertices with d-dimensional features $\mathbf{X}_u$ for each vertex u. For a fixed radius $\ell > 0$, an ℓ-local node-classifier is a function h that takes as input a root vertex $u \in [n]$, the features of all nodes within the ℓ-neighbourhood of u, i.e., $\{\mathbf{X}_v\}_{v\in\eta_\ell(u)}$ and the canonical distances of each node from u, i.e., $\{d(u,v)\}_{v\in\eta_\ell(u)}$; and outputs a classification label for u.

Let $\mathcal{C}_\ell$ denote the class of ℓ-local classifiers. Suppose now that we have a sequence of (random) feature decorated graphs (X_n, G_n) with $|V(G)| = n$. Let u_n denote a uniform at random vertex in G_n. Suppose finally that the rooted feature-decorated graphs (X_n, G_n, u_n) locally weakly converge to (X, G, u). We can then define the notion of asymptotically ℓ-locally Bayes optimal classifiers for this sequence of problems.

Definition 5.2.2. We say that a classifier $h^*_\ell \in \mathcal{C}_\ell$ is the asymptotically ℓ-locally Bayes optimal classifier of the root for the sequence $\{(X_n, G_n, u_n)\}$ if it minimizes the probability of misclassification of the root of the local weak limit, (X, G, u), over the class $\mathcal{C}_\ell$, i.e.,

$$h^*_\ell = \operatorname*{argmin}_{h\in\mathcal{C}_\ell} \mathbf{Pr}\left[h(u, \{\mathbf{X}_v\}_{v\in\eta_\ell(u)}, \{d(u,v)\}_{v\in\eta_\ell(u)}) \neq y_u\right].$$

Before turning to our data model, we note here that the reader may ask how the asymptotically ℓ-locally Bayes optimal classifier compares to the non-asymptotic optimal ℓ-local classifier of the random root, u_n. We show this in an appropriate sense in Theorem 10.

5.3 Optimal Classifier

We are now ready to state our first main result that characterizes the asymptotically ℓ-locally Bayes optimal classifier on the CSBM data described in Section 2.2. For a given graph, let $N_k(u)$ be the set of vertices at a distance of exactly k from node u in the graph. Naturally, $N_0(u) = \{u\}$. Further, denote by $\eta_k(u)$ the k-hop neighbourhood of u, i.e., $\eta_k(u) = \cup_{j=0}^{k} N_k(u)$.

Theorem 7 (Bayes optimal message-passing)**.** *For any* $\ell \geq 1$*, the asymptotically* ℓ*-locally Bayes optimal classifier of the root for the sequence* $(G_n, u_n) \sim \mathrm{CSBM}(n, d, \mathbb{P}, \mathbf{Q})$ *is*

$$h^*_\ell(u, \{\mathbf{X}_v\}_{v\in\eta_\ell(u)}, \{d(u,v)\}_{v\in\eta_\ell(u)}) = \operatorname*{argmax}_{y_u\in[C]} \sum_{v\in\eta_\ell(u)} \log\langle \boldsymbol{\rho}(\mathbf{X}_v), \mathbf{Q}^{d(u,v)}_{y_u}\rangle,$$

where $\{\rho_i\}_{i\in[C]}$ *are the densities associated with the distributions* $\mathbb{P}_i \in \mathbb{P}$*, and* $\boldsymbol{\rho} = (\rho_i)_{i\in[C]}$*, and* $\langle\cdot,\cdot\rangle$ *denotes the vector inner product.*

Let us briefly discuss the meaning of Theorem 7. It states that universally among all ℓ-local classifiers, h^*_ℓ is asymptotically Bayes optimal for the sparse CSBM data. We view $\log\langle \boldsymbol{\rho}(\mathbf{X}_v), \mathbf{Q}^{d(u,v)}_{y_u}\rangle$ as the message gathered from node v that is distance $d(u,v)$ away from node u, where $d(u,u) = 0$ by convention. In particular, $\log\langle \boldsymbol{\rho}(\mathbf{X}_v), \mathbf{Q}^{d(u,v)}_{y_u}\rangle$ computes the alignment of the feature profile of node v in all class $j \in [C]$, $\boldsymbol{\rho}(\mathbf{X}_v)$ with the connectivity profile with node u at distance $d(u,v)$, $\mathbf{Q}^{d(u,v)}_{y_u}$. Furthermore, this optimal classifier is realizable using Architecture 2 (see for example, [LPW$^+$17, Theorem 1], where it is shown that any Lebesgue measurable function can be approximated arbitrarily closely by standard neural networks). Consequently, this result shows that in the sparse setting, the message-passing paradigm can realize the optimal node classification scheme irrespective of the distributions of the node features or the inter-class edge probabilities.

For an intuitive understanding of Theorem 7, it helps to consider two extreme cases. First, if $\mathbf{Q} = p\mathbf{I}$ for some $p \in [0,1]$, then the classifier reduces to a simple convolution,

$$h^*_\ell = \operatorname*{argmax}_{i\in[C]} \left\{ \sum_{v\in\eta_\ell(u)} \log \rho_i(\mathbf{X}_v) \right\}.$$

Second, if $\mathbf{Q} = p\mathbf{1}\mathbf{1}^\top$, then $q_{ij} = p$ for all $i, j \in [C]$, meaning that the graph component of the data is Erdös-Rényi, and hence, completely uninformative for the purposes of node classification. In this case, $\mathbf{Q}_i$ is the same for all i, so the messages are independent of the maximization variable y_u. Hence, the classifier reduces to

$$h^*_\ell = \operatorname*{argmax}_{i\in[C]} \{\log \rho_i(\mathbf{X}_u)\},$$

i.e., it is optimal to look at only the features of node u to predict its label since the neighbourhood does not provide any meaningful information. We formalize this intuition later for a simpler case (see Theorem 9).

5.4 Comparative Study

In this section, we perform a theoretical analysis of the classifier in Theorem 7 using a well-studied specialization of the CSBM data model described in Section 2.2. For ease of discussion, let us restrict ourselves to the setting where there are two classes. Formally, we have $C = 2$, and without loss of generality, the class labels $y_u \in \{\pm 1\}$ for all $u \in [n]$. The distributions of the node features are given by $\mathbf{X}_u \sim \mathbb{P}_{y_u}$ with corresponding density ρ_{y_u}. Furthermore, $\mathbf{Q} = \{q_{ij}\}$ is a 2×2 matrix with $q_{ii} = p = a/n$ and $q_{ij} = q = b/n$ with constants $a > 1, b \geq 0$ for classes $i \neq j$. For a data sample $G = (\mathbf{A}, \mathbf{X})$ from this model, we write $G \sim \mathrm{CSBM}(n, d, \{\mathbb{P}_\pm\}, \mathbf{Q})$ or $G \sim \mathrm{CSBM}(n, d, \{\mathbb{P}_\pm\}, \frac{a}{n}, \frac{b}{n})$. We also recognize the quantity associated with the signal-to-noise ratio (SNR) in the graph structure for this case, which is given by

$$\gamma = \frac{|p - q|}{p + q} = \frac{|a - b|}{a + b}. \tag{5.1}$$

Note that the quantity γ has been recognized as the meaningful SNR in several related works where the underlying random graph model is the binary symmetric stochastic block model, for example, [BFJ21b, FLY+23, WYJ+22, BFJ23a].

Let us now state Theorem 7 in the case of two classes.

Corollary 7.1 (Optimal classifier for binary symmetric CSBM)**.** *For any $\ell \geq 1$, the asymptotically ℓ-locally Bayes optimal classifier of the root for the sequence $(G_n, u_n) \sim \mathrm{CSBM}(n, d, \mathbb{P}, \frac{a}{n}, \frac{b}{n})$ is*

$$h^*_\ell(u, \{\mathbf{X}_v\}_{v \in \eta_\ell(u)}, \{d(u,v)\}_{v \in \eta_\ell(u)}) = \mathrm{sgn}\left(\sum_{v \in \eta_\ell(u)} \mathcal{M}(\mathbf{X}_v, d(u,v)) \right),$$

where $\psi(\mathbf{x}) = \frac{\rho_+(\mathbf{x})}{\rho_-(\mathbf{x})}$ and $\mathcal{M}(\mathbf{x}, k) = \log\left(\frac{1-\gamma^k+\psi(\mathbf{X}_v)(1+\gamma^k)}{1+\gamma^k+\psi(\mathbf{X}_v)(1-\gamma^k)}\right)$.

In this simplified setting, we note that the messages propagated from nodes in the ℓ-local neighbourhood of node u are scaled proportional to a function of their distance k from node u. In particular, this scaling can be expressed in terms of the graph SNR γ from (5.1). It is interesting to observe that $\mathcal{M}_k$ decreases rapidly as k increases. Since $\gamma < 1$, this means that to predict the label of node u, the importance of the information from node v at distance k from u decreases as k increases.

The above simplification helps us interpret the classifier in terms of the graph SNR γ. We will now impose an assumption on the distribution of node features. This will help us analyze the generalization error in terms of the SNR in both the features and the graph, and enable us to compare the performance with other learning methods that are well-studied in the same statistical settings. We will resort to the setting where the features of the CSBM follow a Gaussian mixture. Note that this specialized statistical model has been studied extensively in previous works for benchmarking existing GNN architectures, see for example, [BFJ21b, FLY+23, WYJ+22, BFJ23a].

We report the generalization error for Gaussian features. The generalization error is defined for a classifier h to be the probability of disagreement between the true label y_u and the output of the classifier h_u for node u. We characterize the error for h^*_ℓ in the case where $\mathbb{P}_-, \mathbb{P}_+$ correspond to the Gaussian mixture with components $\mathcal{N}(-\boldsymbol{\mu}, \sigma^2\mathbf{I})$ and $\mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbf{I})$ for fixed $\boldsymbol{\mu} \in \mathbb{R}^d$ and $\sigma > 0$.

In this case, a notion of the signal-to-noise ratio of the features naturally exists, i.e., $\zeta = \|\boldsymbol{\mu}\|_2/\sigma$, a quantity proportional to the ratio of the distance between the means of the mixture and the standard deviation. The log-likelihood ratio in this setting is $\psi(\mathbf{x}) = \log\frac{\rho_+(\mathbf{x})}{\rho_-(\mathbf{x})} = \frac{2}{\sigma^2}\langle \mathbf{x}, \boldsymbol{\mu}\rangle$.

Consider a sequence $\{(G_n, u_n)\}_{n\ge 1}$ with $G_n = (V(G_n), E(G_n))$ from this model where $u_n \sim \mathrm{Unif}(V(G_n))$. In this setting, in the absence of features, it is known that (G_n, u_n) converges locally weakly to a Poisson Galton-Watson tree (see for example, [MNS15b, Section 4]). Here, for every node, we additionally have features that are independent of the graph, and hence, as a straightforward consequence of [MNS15b, Section 4], (G_n, u_n) in our case converges to a feature-decorated Poisson Galton-Watson tree (G, u).

For the root node u, let α_k and β_k denote the number of children at generation k in class y_u and $-y_u$ respectively, where y_u denotes the label of node u. Then $\{\alpha_k\}_{k\ge 0}$ and $\{\beta_k\}_{k\ge 0}$ are characterized by

$$\begin{aligned}
&\alpha_0 = 1, \beta_0 = 0,\\
&\alpha_k \sim \mathrm{Poi}\left(\frac{a\alpha_{k-1} + b\beta_{k-1}}{2}\right), \beta_k \sim \mathrm{Poi}\left(\frac{a\beta_{k-1} + b\alpha_{k-1}}{2}\right) \text{ for } k \in [\ell].
\end{aligned} \tag{5.2}$$

For a classifier h acting on G, let $\mathcal{E}(h)$ denote the probability of misclassification of the root u in G, i.e., $\mathcal{E}(h) = \mathbf{Pr}(h_u y_u < 0)$. Correspondingly, in the case of finite n, we denote by $\mathcal{E}_n(h)$ the probability of misclassification of a uniform random node u_n in G_n. We are now ready to state the generalization error of h^*_ℓ.

Theorem 8 (Generalization error (population risk))**.** *For any $\ell \ge 1$, the generalization error of the asymptotically ℓ-locally Bayes optimal classifier of the root for the sequence*

$(G_n, u_n) \sim \mathrm{CSBM}(n, d, \mathbb{P}, \mathbf{Q})$ *with Gaussian features is given by*

$$\mathcal{E}(h_\ell^*) = \mathbf{Pr}\left[g + \frac{1}{2\zeta}\sum_{k\in[\ell]}\left(\sum_{i\in[\alpha_k]} Z_{k,i}^{(a)} + \sum_{i\in[\beta_k]} Z_{k,i}^{(b)}\right) > \zeta\right],$$

where α_k, β_k *are as in* (5.2), $Z_{k,i}^{(a)} = \varphi(-2\zeta^2 + 2\zeta g_{k,i}, c_k)$, $Z_{k,i}^{(b)} = \varphi(2\zeta^2 + 2\zeta g_{k,i}, c_k)$, *and* $g, \{g_{k,i}\}$ *are mutually independent standard Gaussian random variables.*

Let us now understand how the error described in Theorem 8 behaves in terms of the two SNRs ζ (for the features), and γ (for the graph). Note that $\mathcal{E}(h_\ell^*) \to 0$ as $\zeta \to \infty$, and $\mathcal{E}(h_\ell^*) \to 1/2$ as $\zeta \to 0$. This means that if the signal in the features is large, the number of mistakes made by the classifier vanishes, while if the signal is very small, then roughly half of the nodes are misclassified (equivalent to making a uniform random guess for each node).

To see how γ affects the error, we begin by looking at two extreme settings: first, where the graph is complete noise, i.e., $\gamma = 0$, and second, where the graph signal is very strong, i.e., $\gamma \to 1$, followed by a discussion on how h_ℓ^* interpolates between these extremes. Let $\phi_\pm$ denote the Gaussian density functions with means $\pm\boldsymbol{\mu}$ and variance $\sigma^2\mathbf{I}_d$. Define the random variable

$$\xi_\ell = \xi_\ell(a, b) = \frac{1 + \sum_{k=1}^{\ell} |\alpha_k - \beta_k|}{\sqrt{1 + \sum_{k=1}^{\ell}(\alpha_k + \beta_k)}}, \tag{5.3}$$

where α_k, β_k follow (5.2). In the following, we denote the vanilla GCN classifier from [KW17] by h_{gcn}. We then have the following result.

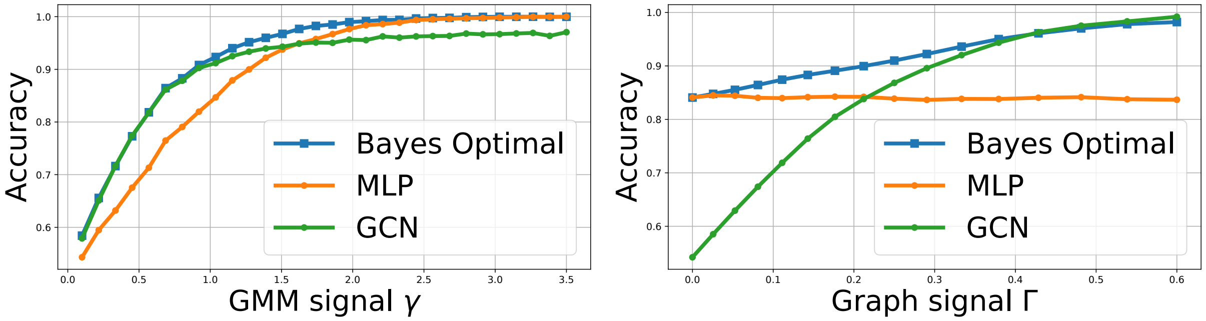
Theorem 9 (Extreme graph signals)**.** *Let* h_ℓ^* *be the classifier described in Corollary 7.1,* $h_0^*(u) = \mathrm{sgn}(\langle \mathbf{X}_u, \boldsymbol{\mu}\rangle)$ *be the Bayes optimal classifier given only the feature information of the root node* u*, and* h_{gcn} *be the one-layer vanilla GCN classifier. Then we have that for any fixed* ℓ*:*

1. *If* $\gamma = 0$ *then* $\mathcal{E}(h_\ell^*) = \mathcal{E}(h_0^*) = \Phi(-\zeta)$*, where* Φ *is the standard Gaussian CDF.*
2. *If* $\gamma \to 1$ *then* $\xi_\ell \geq 1$ *almost surely (a.s.) and* $\mathcal{E}(h_\ell^*) \to \mathbf{Pr}(g > \zeta\xi_\ell)$*, where* $g \sim \mathcal{N}(0, 1)$*.*
3. $\mathcal{E}(h_{\mathrm{gcn}}) = \mathbf{Pr}\,(g > \zeta\xi_1)$*.*

Theorem 9 shows that in the regime of low graph SNR, the optimal classifier h_ℓ^* reduces to a linear classifier $h_0^*(u) = \operatorname{sgn}(\langle \mathbf{X}_u, \boldsymbol{\mu} \rangle)$, which can be realized by a simple MLP that does not use the graph component of the data at all. On the other hand, in the regime of strong graph SNR, h_ℓ^* reduces to a simple convolution over all nodes in the ℓ-neighbourhood and is comparable to a typical GCN. Furthermore, we note that in the strong graph SNR regime $\gamma \to 1$, $\mathcal{E}(h_\ell^*) \to \mathbf{Pr}(g > \zeta\xi_\ell) \leq \Phi(-\zeta)$ since $\xi_\ell \geq 1$. The clip operation during the propagation of messages makes things interesting between these two extremes, where h_ℓ^* interpolates between a simple MLP and an ℓ-hop convolutional network. This interpolation is characterized by the graph signal γ, since the messages are clipped in the range $[-c_k, c_k]$, where $c_k = \log\left(\frac{1+\gamma^k}{1-\gamma^k}\right)$.

In addition, Theorem 9 concludes that if $\xi_1(a, b) > 1$, then a GCN can perform better than every classifier that does not see the graph. On the other hand, if $\xi_1(a, b) < 1$, then a GCN incurs more errors on the data than the best methods that do not use the graph. Interestingly, but not surprisingly, this result aligns with the Kesten-Stigum weak recovery threshold for the community-detection problem on the sparse stochastic block model [Mas14, MNS18], meaning that if weak recovery is possible on the graph component of the data, then a GCN is able to exploit it to perform better than methods that do not use the graph, e.g., a simple MLP.

We now demonstrate our results through experiments using `pyg` (torch-geometric) [FL19]. The following simulations are for the setting $n = 10000$ and $d = 4$ for binary classification on the CSBM. We implement Architecture 2 for the binary case, and perform full-batch training on a graph sampled from the CSBM with certain signals (mentioned in the figures), followed by an evaluation of the architecture on a new graph sampled from the same distribution.



(a) Varying ζ with fixed $\gamma = 0.42$.

(b) Varying γ with fixed $\zeta = 1$.

Figure 5.1: Comparison of Architecture 2 against an MLP and a vanilla GCN.

In Fig. 5.1, we show that the accuracy obtained by the optimal classifier is higher than both a simple MLP and a vanilla GCN [KW17]. We plot the test accuracy of Architecture 2 against the SNR in the node features, $\zeta = \|\boldsymbol{\mu}\| / \sigma$ in Fig. 5.1a, and against the graph SNR $\gamma = |a - b|/(a + b)$ in Fig. 5.1b. We fix $\gamma = 0.42$ and $\zeta = 1$ for the two plots, respectively. We chose these specific values because they generate relatively clearer plots where the accuracy metrics for the three architectures are easily visible and distinguished from each other. The results are similar for other values for γ and ζ, i.e., the Bayes optimal architecture is superior to both MLP and GCN.

Finally, we observe that for the binary setting when the parameters of the architecture are initialized uniformly at random, gradient descent converges and the neural network learns the right parameters such that Architecture 2 realizes the optimal classifier in Corollary 7.1.

5.5 Non-asymptotic Setting

We now turn to the non-asymptotic regime and argue that for fixed n, the classifier in Corollary 7.1 is still in a formal sense, Bayes optimal for an overwhelming fraction of nodes. We begin by exploiting the fact that for up to logarithmic depth neighbourhoods, a sparse CSBM graph is tree-like.

In particular, Proposition 5.6.4 states that for $\ell = c \log n$ for a suitable constant c, the ℓ-neighbourhood of an overwhelming fraction of nodes is a tree. This implies that the classifier h_ℓ^* is Bayes optimal for roughly all of the nodes. Moreover, since the diameter of a sparse graph (as in our setting) is $O(\log n)$ a.s. [CL01, Theorem 6], any learning mechanism can only look as far as $O(\log n)$-hops away from a node to gather new information. This shows that for such graphs, GNNs that are not very deep and look at only up to logarithmic distance in the neighbourhood are sufficient.

Let us now turn to the misclassification error in the non-asymptotic setting. Recall that for a classifier $h \in \mathcal{C}_\ell$, we denote by $\mathcal{E}_n(h)$ and $\mathcal{E}(h)$ the misclassification error of h on the data model with n nodes, and on the limiting data model with $n \to \infty$, respectively. Furthermore, recall from Corollary 7.1 that $\min_{h \in \mathcal{C}_\ell} \mathcal{E}(h) = \mathcal{E}(h_\ell^*)$. Our next result shows that the optimal misclassification error in the non-asymptotic setting across all ℓ-local classifiers, i.e., $\min_{h \in \mathcal{C}_\ell} \mathcal{E}_n(h)$, is close to the misclassification error obtained in the non-asymptotic setting by h_ℓ^*. Moreover, $\min_{h \in \mathcal{C}_\ell} \mathcal{E}_n(h)$ is also close to $\mathcal{E}(h_\ell^*)$ which is explicitly computed in Theorem 8.

Theorem 10 (Misclassification error for fixed n)**.** *For any $1 \leq \ell \leq c \log n$ such that the positive constant c satisfies $c \log(\frac{a+b}{2}) < 1/4$, we have that*

$$\left| \min_{h \in \mathcal{C}_\ell} \mathcal{E}_n(h) - \mathcal{E}_n(h_\ell^*) \right| = O\left(\frac{1}{\log^2 n} \right), \quad \left| \min_{h \in \mathcal{C}_\ell} \mathcal{E}_n(h) - \mathcal{E}(h_\ell^*) \right| = O\left(\frac{1}{\log^2 n} \right).$$

Recall that Corollary 7.1 implies that h_ℓ^* performs optimally on the limiting data model (asymptotic setting) among the class of ℓ-local classifiers $\mathcal{C}_\ell$, but it may not be optimal for the non-asymptotic data model where we have a finite feature-decorated graph with n nodes. However, Theorem 10 helps us conclude that even in the non-asymptotic setting, h_ℓ^* performs almost as well as the actual optimal classifier among $\mathcal{C}_\ell$ in this case, as long as we compare with classifiers that can only look at moderate logarithmic depths in the local neighbourhood, i.e., $\ell \leq c \log n$ for a suitable c.

5.6 Proofs of Results

In this section, we will look at the proof of the results in this chapter. First, I will state two important preliminary results and a fact that are used to establish all our results.

Lemma 5.6.1. *[DKLX+18, Lemma 3]. Let $G \sim \mathrm{CSBM}(n, d, \{\mathbb{P}_-, \mathbb{P}_+\}, \frac{a}{n}, \frac{b}{n})$ and r be a fixed constant. Then the probability that there exists an r-neighbourhood in G with m more edges than the number of vertices is bounded as follows:*

$$\mathbf{Pr}(\exists\, G_r \subset G \text{ s.t. } |E(G_r)| \geq |V(G_r)| + m) \leq \frac{(2(r(m+1))^2(a+b))^{2r(m+1)+m}}{n^m}.$$

Lemma 5.6.2. *[Mas14, Lemma 4.2]. Assume $\ell = c \log(n)$ with $c \log \Delta < 1/4$, where $\Delta = (a+b)/2$. Then with high probability, no node i has more than one cycle in its ℓ-neighbourhood. Moreover, for any $m > 0$, with probability at least $1 - O(1/m^2)$ the number of nodes i whose ℓ-neighbourhood contains at least one cycle is bounded by $O(m \log^3(n) \Delta^{2\ell})$.*

5.6.1 Proof of Theorem 7

In this section, we compute the asymptotically ℓ-locally Bayes optimal classifier for the general CSBM described in Section 2.2 and establish Theorem 7, followed by a proof of Corollary 7.1. Next, we compute the generalization error for the two-class case with arbitrary node features.

For the proofs, we introduce the notation $N_k(u)$ for a given graph to mean the set of vertices at a distance of exactly k from node u in the graph. Thus, $\eta_k(u) = \cup_{j=0}^{k} N_k(u)$. We begin by writing the MAP estimation for this problem. Note that the features $\mathbf{X}_v \in \mathbb{R}^d$ for every node $v \in [n]$ follow the law $\mathbb{P}_i$ if $y_v = i \in [C]$. In addition, recall that we have the edge-probability matrix $\mathbf{Q} = \{q_{ij}\}_{i,j\in[C]}$ with $\mathbf{Pr}((u,v) \text{ is an edge} \mid y_u = i, y_v = j) = q_{ij}$, where $q_{ij} = b_{ij}/n$ for absolute constants b_{ij}. Then we can write the likelihood of the ℓ-neighbourhood of node u as the joint function:

$$f_{\mathbf{Y}}(u, \eta_\ell(u), \{\mathbf{X}_v\}_{v\in\eta_\ell(u)}, \{d(u,v)\}_{v\in\eta_\ell(u)}, \mathbf{Y}) = \mathbf{P}_{\{y_v\}_{v\in\eta_\ell(u)}} \prod_{v\in\eta_\ell(u)} \rho_{y_v}(\mathbf{X}_v)\mathbf{Q}_{y_u y_v}^{d(u,v)}. \tag{5.4}$$

In the above, $\mathbf{P}_{\{y_v\}_{v\in\eta_\ell(u)}}$ denotes the prior distribution of the node labels, which by our assumption is uniform; $\mathbf{Q}_{ij}^k$ denotes the i,j-th entry of the matrix $\mathbf{Q}^k$ and quantifies the probability that a node in class j is at a distance k from a node in class i, and $d(u,v)$ denotes the distance between nodes u and v. Let us now compute the MAP estimator by marginalizing over the labels of all nodes $v \in \eta_\ell(u)\backslash\{u\}$, i.e., $f_{y_u}(u, \eta_\ell(u), \{\mathbf{X}_v\}_{v\in\eta_\ell(u)}, \{d(u,v)\}_{v\in\eta_\ell(u)}, y_u) = \int f_{\mathbf{Y}}\, d\mathbf{Y}_{-u}$, where $\mathbf{Y}_{-u}$ denotes the set of labels other than that of node u.

Let us recall and introduce some notation for the model with C classes. Let the neighbourhood be $\eta_\ell(u) = \{u, v_i, \dots, v_L\}$, where $L = |\eta_\ell(u)| - 1$, the number of nodes in the ℓ-neighbourhood of u other than u itself. Let $V_k = \{v_1, \dots, v_k\}$ for all $1 \le k \le L$ and define

$$T_k = \sum_{\{y_{v_1},\dots,y_{v_k}\}\in[C]^k} \left[\prod_{v\in V_k} \rho_{y_v}(\mathbf{X}_v)\mathbf{Q}_{y_u y_v}^{d(u,v)}\right]. \tag{5.5}$$

We now provide a result to help us compute the maximization of the marginalized likelihood.

Lemma 5.6.3. *Let $i = y_u$ and T_k be defined as in Eq. (5.5). Then we have $T_k = \prod_{v\in V_k}\langle \boldsymbol{\rho}(\mathbf{X}_v), \mathbf{Q}_i^{d(u,v)}\rangle$, where $\boldsymbol{\rho}(\mathbf{x}) \in \mathbb{R}^C$ is the vector of density function outputs $\{\rho_j(\mathbf{x})\}_{j\in[C]}$ and $\mathbf{Q}_i^{d(u,v)}$ denotes the i^{th} row vector of the matrix $\mathbf{Q}^{d(u,v)}$.*

Proof. We will prove this by induction on k. Let $i = y_u$ and $j = y_{v_1}$. Then note that for the base case, we have $T_1 = \sum_{j\in[C]} \rho_j(\mathbf{X}_{v_1})\mathbf{Q}_{ij}^{d(u,v_1)} = \langle \boldsymbol{\rho}(\mathbf{X}_{v_1}), \mathbf{Q}_i^{d(u,v_1)}\rangle$. For the induction hypothesis, assume that $T_l = \prod_{v\in V_l}\langle \boldsymbol{\rho}(\mathbf{X}_v), \mathbf{Q}_i^{d(u,v)}\rangle$ for all $l \le k-1$. Then we have

$$T_k = \sum_{\{y_{v_1},\dots,y_{v_k}\}\in[C]^k} \left[\prod_{v\in V_k} \rho_{y_v}(\mathbf{X}_v)\mathbf{Q}_{y_u y_v}^{d(u,v)}\right]$$

$$
\begin{aligned}
&= \sum_{y_{v_k}\in[C]} \left[\rho_{y_{v_k}}(\mathbf{X}_{v_k})\mathbf{Q}^{d(u,v_k)}_{y_u y_{v_k}} \left(\sum_{\{y_{v_1},\dots,y_{v_{k-1}}\}\in[C]^{k-1}} \left(\prod_{v\in V_k} \rho_{y_v}(\mathbf{X}_v)\mathbf{Q}^{d(u,v)}_{y_u y_v} \right)\right)\right] \\
&= \sum_{y_{v_k}\in[C]} \rho_{y_{v_k}}(\mathbf{X}_{v_k})\mathbf{Q}^{d(u,v_k)}_{y_u y_{v_k}} T_{k-1} && \text{(using Eq. (5.5))} \\
&= \sum_{y_{v_k}\in[C]} \rho_{y_{v_k}}(\mathbf{X}_{v_k})\mathbf{Q}^{d(u,v_k)}_{y_u y_{v_k}} \prod_{v\in V_{k-1}} \langle \boldsymbol{\rho}(\mathbf{X}_v), \mathbf{Q}^{d(u,v)}_i \rangle && \text{(using I.H. to replace } T_{k-1}\text{)} \\
&= \prod_{v\in V_k} \langle \boldsymbol{\rho}(\mathbf{X}_v), \mathbf{Q}^{d(u,v)}_i \rangle.
\end{aligned}
$$

This completes the proof of the lemma. □

Let us now return to the proof of Theorem 7 and continue with the likelihood maximization.

$$
\begin{aligned}
&h^*_\ell(u, \{\mathbf{X}_v\}_{v\in\eta_\ell(u)}, \{d(u,v)\}_{v\in\eta_\ell(u)}) \\
&= \operatorname*{argmax}_{y_u\in[C]} f_{y_u}(u, \{\mathbf{X}_v\}_{v\in\eta_\ell(u)}, \{d(u,v)\}_{v\in\eta_\ell(u)}, y_u) \\
&= \operatorname*{argmax}_{y_u\in[C]} \left\{ \rho_{y_u}(\mathbf{X}_u) \sum_{\{y_{v_1},\dots,y_{v_L}\}\in[C]^L} \left[\prod_{v\in V_L} \rho_{y_v}(\mathbf{X}_v)\mathbf{Q}^{d(u,v)}_{y_u y_v} \right] \right\} \\
&= \operatorname*{argmax}_{y_u\in[C]} \ \rho_{y_u}(\mathbf{X}_u)\, T_L && \text{using Eq. (5.5)} \\
&= \operatorname*{argmax}_{y_u\in[C]} \ \rho_{y_u}(\mathbf{X}_u) \prod_{v\in V_L} \langle \boldsymbol{\rho}(\mathbf{X}_v), \mathbf{Q}^{d(u,v)}_{y_u} \rangle && \text{using Lemma 5.6.3} \\
&= \operatorname*{argmax}_{y_u\in[C]} \ \log(\rho_{y_u}(\mathbf{X}_u)) + \sum_{v\in V_L} \log\langle \boldsymbol{\rho}(\mathbf{X}_v), \mathbf{Q}^{d(u,v)}_{y_u} \rangle.
\end{aligned}
$$

Here, $\log\langle \boldsymbol{\rho}(\mathbf{X}_v), \mathbf{Q}^{d(u,v)}_{y_u} \rangle$ is seen as the information obtained from a node v in $\eta_\ell(u)$ which is at a distance $d(u,v)$ from u, to estimate the label of u. Furthermore, note that an instance of Architecture 2 with $L = 1$, $\sigma_1 = \{\rho_i\}_{i\in[C]}$, and $\mathbf{Q}$ for the edge-probabilities realizes the function h^*_ℓ for a given root node u and its ℓ-neighbourhood $\eta_\ell(u)$, in the sense that $\hat{y}_u = h^*_\ell(u, \{\mathbf{X}_v\}_{v\in\eta_\ell(u)}, \{(d(u,v))\}_{v\in\eta_\ell(u)})$. This completes the proof of Theorem 7.

Next, we obtain a simpler version of the classifier for the two-class symmetric CSBM with an arbitrary distribution of node features. Recall that ψ is the likelihood ratio. The proof follows directly from Theorem 7, by taking $C = 2$. In this two-class case, the features $\mathbf{X}_v \in \mathbb{R}^d$ for every node follow the law $\mathbb{P}_{y_v}$ where the class labels $y_v \in \{\mp 1\}$ are $\mathrm{Unif}(\{-1,+1\})$. In addition, we have

$$\mathbf{Pr}((u,v) \text{ is an edge}) = \begin{cases} p & y_u = y_v \\ q & y_u \neq y_v \end{cases},$$

where $p = a/n$ and $q = b/n$ for absolute constants $a > 1, b \geq 0$. Define the quantities p_k, q_k as follows for $k \in [\ell]$:

$$p_k = \sum_{j=0}^{\lfloor k/2 \rfloor} \binom{k}{2j} p^{k-2j} q^{2j}, \qquad q_k = \sum_{j=1}^{\lceil k/2 \rceil} \binom{k}{2j-1} p^{k-2j+1} q^{2j-1}. \tag{5.6}$$

Now note that $r_k = \frac{p_k}{q_k} = \frac{(p+q)^k + (p-q)^k}{(p+q)^k - (p-q)^k} = \frac{1+\gamma^k}{1-\gamma^k}$, where γ is given in Eq. (5.1). Then the classifier reduces to

$$\begin{aligned}
&h^*_\ell(u, \{\mathbf{X}_v\}_{v \in \eta_\ell(u)}, \{d(u,v)\}_{v \in \eta_\ell(u)}) \\
&= \operatorname*{argmax}_{y_u \in \{\pm 1\}} \sum_{v \in \eta_\ell(u)} \log \langle \boldsymbol{\rho}(\mathbf{X}_v), \mathbf{Q}^{d(u,v)}_{y_u} \rangle \\
&= \operatorname*{argmax}_{y_u \in \{\pm 1\}} \log(\rho_{y_u}(\mathbf{X}_u)) + \sum_{v \in \eta_\ell(u) \setminus \{u\}} \log \langle \boldsymbol{\rho}(\mathbf{X}_v), \mathbf{Q}^{d(u,v)}_{y_u} \rangle \\
&= \operatorname*{argmax}_{y_u \in \{\pm 1\}} \log(\rho_{y_u}(\mathbf{X}_u)) + \sum_{k \in [\ell]} \sum_{v \in N_k(u)} \log \langle \boldsymbol{\rho}(\mathbf{X}_v), \mathbf{Q}^{k}_{y_u} \rangle \\
&= \operatorname*{argmax}_{y_u \in \{\pm 1\}} \log(\rho_{y_u}(\mathbf{X}_u)) + \sum_{k \in [\ell]} \sum_{v \in N_k(u)} \log \left(\rho_+(\mathbf{X}_v) p_k^{1+y_u/2} q_k^{1-y_u/2} + \rho_-(\mathbf{X}_v) p_k^{1-y_u/2} q_k^{1+y_u/2} \right) \\
&= \operatorname{sgn}\left(\log \frac{\rho_+(\mathbf{X}_u)}{\rho_-(\mathbf{X}_u)} + \sum_{k \in [\ell]} \sum_{v \in N_k(u)} \log \left(\frac{\rho_+(\mathbf{X}_v) p_k + \rho_-(\mathbf{X}_v) q_k}{\rho_+(\mathbf{X}_v) q_k + \rho_-(\mathbf{X}_v) p_k} \right) \right) \\
&= \operatorname{sgn}\left(\log \psi(\mathbf{X}_u) + \sum_{k \in [\ell]} \sum_{v \in N_k(u)} \log \left(\frac{1 + r_k \psi(\mathbf{X}_v)}{r_k + \psi(\mathbf{X}_v)} \right) \right) \\
&= \operatorname{sgn}\left(\log \psi(\mathbf{X}_u) + \sum_{k \in [\ell]} \sum_{v \in N_k(u)} \log \left(\frac{1 - \gamma^k + \psi(\mathbf{X}_v)(1+\gamma^k)}{1 + \gamma^k + \psi(\mathbf{X}_v)(1-\gamma^k)} \right) \right) \qquad \left(\text{using } r_k = \frac{1+\gamma^k}{1-\gamma^k} \right) \\
&= \operatorname{sgn}\left(\sum_{k=0}^{\ell} \sum_{v \in N_k(u)} \log \left(\frac{1 - \gamma^k + \psi(\mathbf{X}_v)(1+\gamma^k)}{1 + \gamma^k + \psi(\mathbf{X}_v)(1-\gamma^k)} \right) \right)
\end{aligned}$$

$$= \operatorname{sgn}\left(\sum_{v\in\eta_\ell(u)} \log\left(\frac{1-\gamma^{d(u,v)}+\psi(\mathbf{X}_v)(1+\gamma^{d(u,v)})}{1+\gamma^{d(u,v)}+\psi(\mathbf{X}_v)(1-\gamma^{d(u,v)})}\right)\right) = \operatorname{sgn}\left(\sum_{v\in\eta_\ell(u)} \mathcal{M}(\mathbf{X}_v, d(u,v))\right)$$

5.6.2 Proof of Theorem 8

Let us now compute the generalization error of h_ℓ^*. Formally, given a data instance $(u, \{\mathbf{X}_v\}_{v\in\eta_\ell(u)})$ along with the neighbourhood $\eta_\ell(u)$, h_ℓ^* outputs a label $\hat{y}_u \in \{\pm 1\}$, and the generalization error is defined as the probability $\mathbf{Pr}(y_u\hat{y}_u < 1)$. For a simple calculation, let us assume that the latent labels y_i are uniformly distributed, i.e., $\mathbf{Pr}(y_i = -1) = \mathbf{Pr}(y_i = 1) = \frac{1}{2}$. It is straightforward to generalize to unbalanced settings. Recall that the features in classes ±1 follow the law $\mathbb{P}_\pm$, and denote the likelihood ratio by $\psi(\mathbf{x}) = \rho_+(\mathbf{x})/\rho_-(\mathbf{x})$. Then we have that for a fixed u,

$$\begin{aligned}\mathcal{E}(h_\ell^*) &= \mathbf{Pr}\left(y_u\left(\log\psi(\mathbf{X}_u) + \sum_{k\in[\ell]}\sum_{v\in N_k(u)} \mathcal{M}_k(\mathbf{X}_v)\right) < 0\right)\\ &= \frac{1}{2}\left[\mathbf{Pr}\left(\log\psi(\mathbf{Y}^{(1)}) + \sum_{k\in[\ell]} \mathbf{Z}_k^{(1)} > 0\right) + \mathbf{Pr}\left(\log\psi(\mathbf{Y}^{(2)}) + \sum_{k\in[\ell]} \mathbf{Z}_k^{(2)} < 0\right)\right],\end{aligned}$$

where $\mathbf{Y}^{(1)} \sim \mathbb{P}_-$, $\mathbf{Y}^{(2)} \sim \mathbb{P}_+$, $\mathbf{Z}_k^{(1)} = \sum_{j\in[\alpha_k]} \mathcal{M}_k(\mathbf{Y}_{k,j}^{(1)}) + \sum_{j\in[\beta_k]} \mathcal{M}_k(\mathbf{Y}_{k,j}^{(2)})$ and $\mathbf{Z}_k^{(2)} = \sum_{j\in[\alpha_k]} \mathcal{M}_k(\mathbf{Y}_{k,j}^{(2)}) + \sum_{j\in[\beta_k]} \mathcal{M}_k(\mathbf{Y}_{k,j}^{(1)})$ are independent random variables with $\mathbf{Y}_{k,j}^{(1)} \sim \mathbb{P}_-$ and $\mathbf{Y}_{k,j}^{(2)} \sim \mathbb{P}_+$.

The above expression is not particularly insightful. For this reason, we now specialize to the case of Gaussian features and interpret the error in terms of the natural SNRs associated with the Gaussian mixture and the graph, i.e., the quantities ζ and γ.

For the Gaussian mixture, the log of the likelihood ratio for a node u is given by $\frac{2}{\sigma^2}\langle \mathbf{X}_u, \boldsymbol{\mu}\rangle \overset{\mathcal{D}}{=} 2y_u\zeta^2 + 2\zeta g$, where $g \sim \mathcal{N}(0,1)$. Replacing every $\{\mathbf{X}_i\}_{i\in[n]}$ as $\mathbf{X}_i = \mathbb{E}\mathbf{X}_i + \sigma\mathbf{g}_i = y_i\boldsymbol{\mu} + \sigma\mathbf{g}_i$, we obtain the expression in Theorem 8.

5.6.3 Proof of Theorem 9

Let us now turn to the next result, where we analyze the generalization error in the cases where the graph SNR γ takes extreme values.

Note that when $\gamma = 0$, i.e., when $a = b$, then $a_k = b_k$ for all $k \in [\ell]$, hence, $c_k = \log(a_k/b_k) = 0$. This implies that all information from the k-hop neighbours is truncated to 0 for all $k \in [\ell]$. Thus, the classifier reduces to $h^*_u = h^*_\ell(u, \{\mathbf{X}_v\}_{v\in\eta_\ell(u)}) = \mathrm{sgn}\left(\psi(\mathbf{X}_u)\right) = \mathrm{sgn}\left(g + y_u\zeta\right) = h^*_0(u, \{\mathbf{X}_u\})$. Thus, the probability that $h^*_u y_u < 0$ is

$$\mathbf{Pr}(h^*_u y_u < 0) = \mathbf{Pr}(y_u g + \zeta < 0) = \Phi(-\zeta),$$

where $\Phi(\cdot)$ is the standard Gaussian CDF.

For the other case where $\gamma \to 1$, we have two sub-cases: Either $a \to 0$ with $b \neq 0$, or $a \neq 0$ with $b \to 0$. In this case, $c_k \to \infty$ for all k, so the classifier takes the form

$$h^*_\ell(u, \{\mathbf{X}_v\}_{v\in\eta_\ell(u)}) = \mathrm{sgn}\left(g + y_u\zeta + \sum_{k\in[\ell]}\sum_{v\in N_k(u)}(g_{k,v} + y_v\zeta)\right).$$

Hence, the probability of making a mistake is

$$\begin{aligned}\mathbf{Pr}(h^*_u y_u < 0) &= \mathbf{Pr}\left(y_u g + \zeta + \sum_{k\in[\ell]}\sum_{v\in N_k(u)}(y_u g_{k,v} + y_u y_v\zeta) < 0\right)\\ &= \mathbf{Pr}\left(g > \zeta\frac{|\eta^{(a)}_\ell - \eta^{(b)}_\ell|}{\sqrt{\eta^{(a)}_\ell + \eta^{(b)}_\ell}}\right),\end{aligned}$$

where $\eta^{(a)}_\ell, \eta^{(b)}_\ell$ denote the total number of nodes in the ℓ-neighbourhood $\eta_\ell(u)$ that are in the same class as u and different class as u, respectively. The last equation is obtained by using the fact that $(g, \{g_{k,v}\})$ are iid standard Gaussians. Note that in this case since either $b \to 0$ or $a \to 0$ (but not both), we have $\eta^{(b)}_\ell \to 0$ or $\eta^{(a)}_\ell \to 0$ using (5.2) for any fixed ℓ. Thus, $\xi_\ell(a, b) > 1$ a.s. Following a similar analysis, one can find that $\mathcal{E}(h_{\mathrm{gcn}}) = \mathbf{Pr}(g > \zeta \cdot \xi_1(a, b))$. This completes the proof.

It is interesting to note that we may not have $\xi_1(a, b) > 1$ in general, meaning that a GCN is better than methods that do not use a graph only in the case where $\xi_1(a, b) > 1$.

5.6.4 Proof of Theorem 10

First, consider the case where ℓ, the total depth of the neighbourhood is a constant independent of n, the number of nodes.

Putting $m = 1$ in Lemma 5.6.1, we see that the probability is bounded by $(8\ell^2(a+b))^{4\ell+1}/n = O(1/n)$. Hence, we conclude that in the limit $n \to \infty$, there are no cycles in any constant-depth neighbourhoods in the graph. In particular, we obtain that the local weak limit (G, u) is a tree.

We now turn to the case where the depth of the neighbourhood is logarithmic in n.

Proposition 5.6.4 (Tree neighbourhoods)**.** *Let* $G = (V, E) \sim \mathrm{CSBM}(n, d, \mathbb{P}, \frac{a}{n}, \frac{b}{n})$ *for constants* $a, b > 1$*. Then for any* $\ell = c \log n$ *such that* $c \log((a+b)/2) < 1/4$*, with probability* $1 - O(1/\log^2 n)$*, the number of nodes* $u \in V$ *whose* ℓ*-neighbourhood is cycle-free is* $n(1 - o(\log^4(n)/\sqrt{n}))$*.*

Proof. In Lemma 5.6.2, observe that since $c \log \Delta < 1/4$, we have $\ell < \frac{\log n}{4 \log \Delta} = \frac{1}{4} \log_\Delta n$. Thus, putting $m = \log n$, we find that with probability at least $1 - O(1/\log^2 n)$, the number of nodes whose ℓ-neighbourhood contains at least one cycle is bounded by $O(\log^4(n)\Delta^{2\ell}) = o(\log^4(n)\sqrt{n})$. Hence, the fraction of nodes whose ℓ-neighbourhood is cycle-free is $1 - o(\frac{\log^4 n}{\sqrt{n}})$. □

For a fixed node $u \in [n]$, let us denote the number of nodes at distance k (respectively $\leq k$) from u with class label $\pm y_u$ by $U_k^{\pm}(u)$ (respectively, $U_{\leq k}^{\pm}(u)$). Also let $n_\pm$ denote the number of nodes with class label $\pm y_u$, so that $n = n_+ + n_-$. Note that $U_0^+(u) = 1$, $U_0^-(u) = 0$, and conditionally on the sigma-field $\mathcal{F}_{k-1} = \sigma(U_t^{\pm}(u), t \leq k-1)$, we have

$$U_k^+(u) \sim \mathrm{Bin}\left(n_+ - U_{\leq k-1}^+, 1 - (1 - a/n)^{U_{k-1}^+}(1 - b/n)^{U_{k-1}^-}\right), \tag{5.7}$$

$$U_k^-(u) \sim \mathrm{Bin}\left(n_- - U_{\leq k-1}^-, 1 - (1 - a/n)^{U_{k-1}^-}(1 - b/n)^{U_{k-1}^+}\right). \tag{5.8}$$

Define $S_k(u) = U_k^+(u) + U_k^-(u)$ to be the number of nodes at distance exactly k from u, and denote $\Delta = \frac{a+b}{2}$ to be the expected degree of a node. Correspondingly, recall from (5.2) that we have

$$\begin{aligned} &\alpha_0 = 1, \beta_0 = 0, \\ &\alpha_k \sim \mathrm{Poi}\left(\frac{a\alpha_{k-1} + b\beta_{k-1}}{2}\right), \beta_k \sim \mathrm{Poi}\left(\frac{a\beta_{k-1} + b\alpha_{k-1}}{2}\right) \text{ for } k \in [\ell]. \end{aligned} \tag{5.9}$$

Let us now state a useful high-probability bound on $S_k(u) = U_k^+(u) + U_k^-(u)$.

Lemma 5.6.5. *[Mas14, Theorem 2.3]. For any* $\ell = c \log n$ *with* $c \log \Delta < 1/4$*, there exist constants* $C, \epsilon > 0$ *such that with probability at least* $1 - O(n^{-\epsilon})$*,* $S_k(u) \leq C\Delta^k \log(n)$ *for all* $u \in [n]$ *and all* $k \in [\ell]$*.*

We now obtain a total variation bound between the sequences $\{U_k^{\pm}\}_{k\geq 0}$ and $\{\alpha_k, \beta_k\}_{k\geq 0}$.

Lemma 5.6.6. *Let* $u \in [n]$ *be fixed with label* $y_u \in \{\pm 1\}$. *Let* $\ell = c \log n$ *with* $c \log \Delta < 1/4$. *Then the total variation distance between the collections of variables* $\{U_k^+(u), U_k^-(u)\}_{k\leq \ell}$ *and* $\{\alpha_k(u), \beta_k(u)\}_{k\leq\ell}$ *is bounded by* $O(\log^3 n/n^{1/4})$.

Proof. Define the following events for C as in Lemma 5.6.5:

$$\Omega_k = \{S_k \leq C\Delta^k \log n\}, 1 \leq k \leq \ell. \tag{5.10}$$

Conditionally on the sigma-field $\mathcal{F}_{k-1} = \sigma(U_t^{\pm}(u), t \leq k-1)$ and the event Ω_{k-1}, we compute the total variation distance between the variables $(U_k^+(u), U_k^-(u))$ and $(\alpha_k(u), \beta_k(u))$. Since u is fixed, we omit it from the notation for brevity. Define the following random variables:

$$W_k^+ \sim \text{Poi}\left(\frac{aU_{k-1}^+ + bU_{k-1}^-}{2}\right), \qquad W_k^- \sim \text{Poi}\left(\frac{aU_{k-1}^- + bU_{k-1}^+}{2}\right).$$

We now apply the Stein-Chen method to bound $d_{\text{TV}}(U_k^{\pm}, W_k^{\pm})$. For more details on this technique, we refer to [Ste72, Che75, BC05]. In particular, we use the fact that for $X_1 \sim \text{Bin}(n, \lambda/n)$, $X_2 \sim \text{Poi}(\lambda)$ and $X_3 \sim \text{Poi}(\lambda')$, $d_{\text{TV}}(X_1, X_2) \leq \lambda/n$ and $d_{\text{TV}}(X_2, X_3) \leq |\lambda - \lambda'|$. Let us focus on $d_{\text{TV}}(U_k^+, W_k^+)$ as the other case for $d_{\text{TV}}(U_k^-, W_k^-)$ is similar. Construct an intermediate random variable based on the distributions of $U_k^{\pm}$ as in Eqs. (5.7) and (5.8),

$$V_k \sim \text{Poi}\left((n_+ - U_{\leq k-1}^+)\left(1 - (1-a/n)^{U_{k-1}^+}(1-b/n)^{U_{k-1}^-}\right)\right).$$

Denote $T_t = 1 - \left(1 - \frac{a}{n}\right)^{U_t^+}\left(1 - \frac{b}{n}\right)^{U_t^-}$ for brevity. Note that using triangle inequality,

$$\begin{aligned}
d_{\text{TV}}(V_k, W_k^+) &\leq \left|(n_+ - U_{\leq k-1}^+)T_{k-1} - \frac{aU_{k-1}^+ + bU_{k-1}^-}{2}\right| \\
&\leq \left|\left(n_+ - U_{\leq k-1}^+ - \frac{n}{2}\right)T_{k-1}\right| + \left|\frac{aU_{k-1}^+ + bU_{k-1}^- - nT_{k-1}}{2}\right| \\
&= \left|\left(n_+ - U_{\leq k-1}^+ - \frac{n}{2}\right)T_{k-1}\right| + \frac{n}{2}\left|\frac{aU_{k-1}^+ + bU_{k-1}^-}{n} - T_{k-1}\right| \\
&\leq \left|\left(n_+ - U_{\leq k-1}^+ - \frac{n}{2}\right)T_{k-1}\right| + \frac{1}{4n}\left(aU_{k-1}^+ + bU_{k-1}^-\right)^2,
\end{aligned}$$

where in the last inequality we used Fact 2.4.9. Then we obtain the variation distance:

$$\begin{aligned} d_{\mathrm{TV}}(U_k^+, W_k^+) &\le d_{\mathrm{TV}}(U_k^+, V_k) + d_{\mathrm{TV}}(V_k, W_k^+) \\ &\le T_{k-1} + \left| \left(n_+ - U_{\le k-1}^+ - \frac{n}{2} \right) T_{k-1} \right| + \frac{1}{4n} \left(aU_{k-1}^+ + bU_{k-1}^- \right)^2 \\ &= T_{k-1} \left(1 + \left| n_+ - U_{\le k-1}^+ - \frac{n}{2} \right| \right) + \frac{1}{4n} \left(aU_{k-1}^+ + bU_{k-1}^- \right)^2 . \end{aligned}$$

Consider now a choice of c such that $c \log \Delta < 1/4$. We have $\ell = c \log n < \frac{1}{4} \log_\Delta n$, implying that $\Delta^\ell \le n^{1/4}$. Recalling (5.10) corresponding to Lemma 5.6.5, we have that under the event Ω_{k-1} for $k \le \ell$, the number of nodes at distance $k-1$ is

$$S_{k-1} = U_{k-1}^+ + U_{k-1}^- \le C\Delta^{k-1} \log n \le C\Delta^\ell \log n \le Cn^{1/4} \log n. \tag{5.11}$$

Observe now that from Fact 2.4.9, $T_{k-1} \le \frac{aU_{k-1}^+ + bU_{k-1}^-}{n}$. Recalling that y_u have a uniform prior, by the Chernoff bound [Ver18, Theorem 2.3.1] on n_+, we have $|n_+ - \frac{n}{2}| = O(\sqrt{n} \log n)$ with probability at least $1 - 1/\mathrm{poly}(n)$. Thus, we obtain that under this event,

$$\begin{aligned} d_{\mathrm{TV}}(U_k^+, W_k^+) &\le O\left(\frac{|U_{\le k-1}^+|}{n} + \frac{\log n}{\sqrt{n}} \right) \cdot \left(aU_{k-1}^+ + bU_{k-1}^- \right) + \frac{1}{4n} \left(aU_{k-1}^+ + bU_{k-1}^- \right)^2 \\ &\le O\left(\frac{\log n}{\sqrt{n}} \right) \cdot \max(a,b) S_{k-1} + \frac{\max(a,b)^2}{4n} S_{k-1}^2 = O\left(\frac{\log^2 n}{n^{1/4}} \right), \end{aligned}$$

where in the last step we used the bound from (5.11). Now recall that the variables $\{U_k^\pm, \alpha_k, \beta_k\}_{k \in [\ell]}$ are defined as in Eqs. (5.7) to (5.9) for all $k \le \ell$. For a fixed $u \in [n]$, we have the base cases $U_0^+ = \alpha_0 = 1$ and $U_0^- = \beta_0 = 0$. Then following an induction argument with a union bound over all $k \in \{1, \ldots, \ell\}$, we have that the variation distance between the sequences $\{U_k^+, U_k^-\}_{k \le \ell}$ and $\{\alpha_k, \beta_k\}_{k \le \ell}$ is upper bounded by $O\left(\frac{\log^3 n}{n^{1/4}} \right)$. $\square$

We now obtain a relationship between the misclassification error on the data model with finite n, i.e., $\mathcal{E}_n$ and the error on the limit of the model with $n \to \infty$, i.e., $\mathcal{E}$.

Theorem (Restatement of Theorem 10)**.** *For any $1 \le \ell \le c \log n$ such that the positive constant c satisfies $c \log(\frac{a+b}{2}) < 1/4$, we have that*

$$\left| \min_{h \in \mathcal{C}_\ell} \mathcal{E}_n(h) - \mathcal{E}_n(h_\ell^*) \right| = O\left(\frac{1}{\log^2 n} \right), \quad \left| \min_{h \in \mathcal{C}_\ell} \mathcal{E}_n(h) - \mathcal{E}(h_\ell^*) \right| = O\left(\frac{1}{\log^2 n} \right).$$

Proof. Consider a random feature-decorated graph $G_n \sim \mathrm{CSBM}(n, d, \{\mathbb{P}_\pm\}, a/n, b/n)$, where $\mathbb{P}_\pm$ correspond to the distributions $\mathcal{N}(\pm\boldsymbol{\mu}, \sigma^2\mathbf{I})$ for the node features given by $\{\mathbf{X}_u\}_{u\in[n]}$. For a classifier $h \in \mathcal{C}_\ell$, the class of all ℓ-local classifiers, define $\mathcal{E}_n(h)$ to be the probability of misclassification for a uniform at random node $u \in [n]$, i.e., $\mathcal{E}_n(h) = \mathbf{Pr}(y_u \cdot h(u, \{\mathbf{X}_v\}_{v\in\eta_\ell(u)}, \eta_\ell(u)) < 0)$. Since it is known that all classifiers in $\mathcal{C}_\ell$ operate on u given the information in its ℓ-neighbourhood $\eta_\ell(u)$, we will omit $\eta_\ell(u)$ from the notation and say $h(u)$ instead of $h(u, \{\mathbf{X}_v\}_{v\in\eta_\ell(u)}, \eta_\ell(u))$ when it is understood. Let $\mathbf{P}$ be the joint measure of the variables $\{U_k^\pm\}_{k\le\ell}$ from Eqs. (5.7) and (5.8), and $\mathbf{P}'$ be the joint measure of the variables $\{\alpha_k, \beta_k\}_{k\le\ell}$ from Eq. (5.9). Then Lemma 5.6.6 gives us that $d_{\mathrm{TV}}(\mathbf{P}, \mathbf{P}') \le O\left(\frac{\log^3 n}{n^{1/4}}\right) = o_n(1)$.

Recall $\mathcal{E}(h_\ell^*)$ computed in Theorem 8 for the limiting data model (G, u).

$$\begin{aligned}\mathcal{E}(h_\ell^*) &= \mathbf{Pr}\left[g + \frac{1}{2\zeta}\sum_{k\in[\ell]}\left(\sum_{i=1}^{\alpha_k} Z_{k,i}^{(a)} + \sum_{i=1}^{\beta_k} Z_{k,i}^{(b)}\right) > \zeta\right]\\
&= \int \mathbf{Pr}\left[g + \frac{1}{2\zeta}\sum_{k\in[\ell]}\left(\sum_{i=1}^{\alpha_k} Z_{k,i}^{(a)} + \sum_{i=1}^{\beta_k} Z_{k,i}^{(b)}\right) > \zeta \,\middle|\, \{\alpha_k, \beta_k\}_{k\le\ell}\right] d\mathbf{P}'.\end{aligned}$$

Similarly, we have

$$\begin{aligned}\mathcal{E}_n(h_\ell^*) &= \mathbf{Pr}\left[g + \frac{1}{2\zeta}\sum_{k\in[\ell]}\left(\sum_{i=1}^{U_k^+} Z_{k,i}^{(a)} + \sum_{i=1}^{U_k^-} Z_{k,i}^{(b)}\right) > \zeta\right]\\
&= \int \mathbf{Pr}\left[g + \frac{1}{2\zeta}\sum_{k\in[\ell]}\left(\sum_{i=1}^{U_k^+} Z_{k,i}^{(a)} + \sum_{i=1}^{U_k^-} Z_{k,i}^{(b)}\right) > \zeta \,\middle|\, \{U_k^\pm\}_{k\le\ell}\right] d\mathbf{P}.\end{aligned}$$

Thus, we obtain that

$$|\mathcal{E}_n(h_\ell^*) - \mathcal{E}(h_\ell^*)| \le d_{\mathrm{TV}}(\mathbf{P}, \mathbf{P}') \le O\left(\frac{\log^3 n}{n^{1/4}}\right) = o_n(1), \tag{5.12}$$

Let us now focus on the case with finite n. Let A denote the event from Proposition 5.6.4 where the number of nodes with cycle-free ℓ-neighbourhoods is $1 - o(\frac{\log^4 n}{\sqrt{n}})$. For a node $u \in G_n$, let E_u denote the event that the subgraph induced by the ℓ-neighbourhood of u, $\eta_\ell(u)$ is a tree. Then observe that for a uniform random node $u \in G_n$,

$$\min_{h\in\mathcal{C}_\ell} \mathcal{E}_n(h) = \mathbf{Pr}(y_u h_{\ell,n}^*(u) < 0)$$

$$
\begin{aligned}
&= \mathbf{Pr}(E_u)\mathbf{Pr}(y_u h^*_{\ell,n}(u) < 0 \mid E_u) + \mathbf{Pr}(E_u^{\mathsf{c}})\mathbf{Pr}(y_u h^*_{\ell,n}(u) < 0 \mid E_u^{\mathsf{c}}) \\
&= (1 - o_n(1))\mathbf{Pr}(y_u h^*_\ell(u) < 0) + o_n(1) \\
&= \mathcal{E}_n(h^*_\ell) \pm o_n(1).
\end{aligned}
$$

In the above, we used from Proposition 5.6.4 that $\mathbf{Pr}(E_u) = \mathbf{Pr}(E_u \cap A) + \mathbf{Pr}(E_u \cap A^{\mathsf{c}}) = 1 - O(\frac{1}{\log^2 n})$, and that $\mathcal{E}_n(h^*_\ell) = \min_{h\in\mathcal{C}_\ell} \mathbf{Pr}(y_u h(u) < 0 \mid E_u)$. This establishes the first part:

$$
|\min_{h\in\mathcal{C}_\ell} \mathcal{E}_n(h) - \mathcal{E}_n(h^*_\ell)| = O\left(\frac{1}{\log^2 n}\right).
$$

Combining the above display with (5.12), we obtain the second part, i.e.,

$$
\begin{aligned}
\left|\min_{h\in\mathcal{C}_\ell} \mathcal{E}_n(h) - \min_{h\in\mathcal{C}_\ell} \mathcal{E}(h)\right| &= \left|\min_{h\in\mathcal{C}_\ell} \mathcal{E}_n(h) - \mathcal{E}(h^*_\ell)\right| \\
&= \left|\min_{h\in\mathcal{C}_\ell} \mathcal{E}_n(h) - \mathcal{E}_n(h^*_\ell) + \mathcal{E}_n(h^*_\ell) - \mathcal{E}(h^*_\ell)\right| \\
&\le \left|\min_{h\in\mathcal{C}_\ell} \mathcal{E}_n(h) - \mathcal{E}_n(h^*_\ell)\right| + |\mathcal{E}_n(h^*_\ell) - \mathcal{E}(h^*_\ell)| \\
&= O\left(\frac{1}{\log^2 n}\right) + O\left(\frac{\log^3 n}{n^{1/4}}\right) = O\left(\frac{1}{\log^2 n}\right).
\end{aligned}
$$

This completes the proof. □

5.7 Additional Empirical Observations

In this section, I will describe a few empirical observations that may help understand Architecture 2 better in terms of being practically useful, i.e., train-able and comparable to existing graph learning methods. The complete analysis of the implementation and its comparison with other GNN architectures is left as an open problem.

5.7.1 Convergence of Parameters

We observe empirically that gradient descent (`SGD` and `Adam` implementations in the pytorch library) converges in the binary setting. In this case, the neural network learns the correct parameters corresponding to the parameters of the CSBM from which the data is

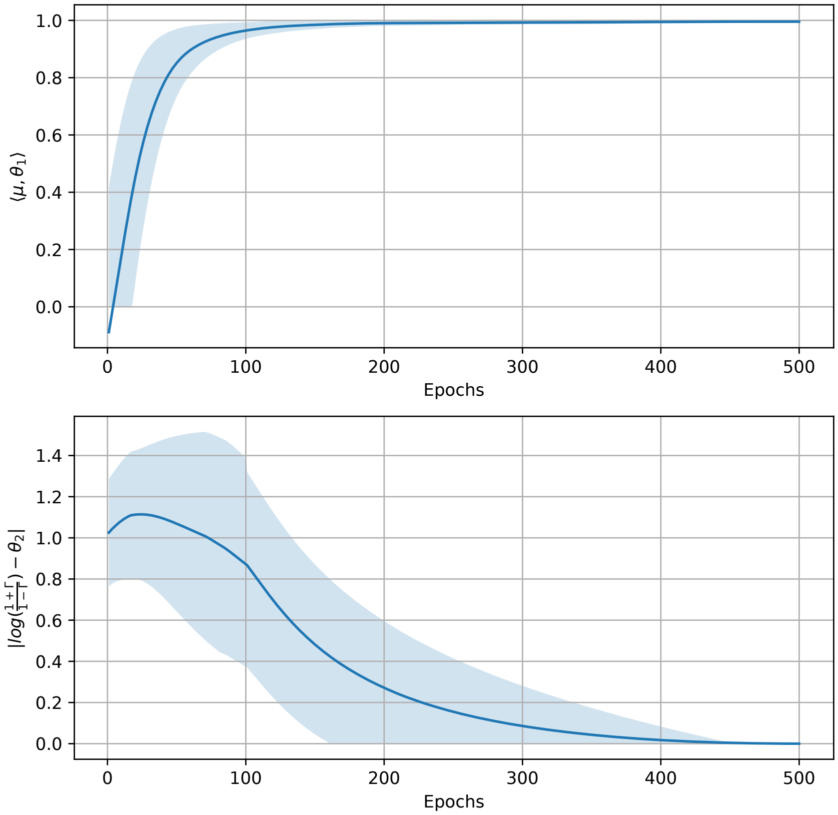


Figure 5.2: Convergence of parameters of Architecture 2.

sampled, i.e., ρ and $\mathbf{Q}$, such that Architecture 2 realizes the optimal classifier in Corollary 7.1. In Fig. 5.2, the x-axis denotes the number of epochs elapsed since the beginning of the training process. For the first plot, the y-axis denotes the cosine similarity between the parameters $\theta_1 \in \mathbb{R}^d$ learned by the MLP $\mathbf{H}^{(L)}$ in Architecture 2 and the ansatz $\boldsymbol{\mu}$ that realizes the optimal classifier; while in the second plot, the y-axis denotes the absolute difference between the clip parameter $\theta_2 \in \mathbb{R}$ and the ansatz value $\log(\frac{1+\gamma}{1-\gamma})$. These experiments are performed in the same setting as Fig. 5.1b with fixed $\gamma = 0.2$. We see that the parameters converge as the number of training iterations increases. The reported metrics are averaged over 10 trials, and the standard deviation is shown at each iteration using the translucent blue region.

5.7.2 Comparison with AMP-BP

We now perform a comparison of our architecture with $\ell = 5$ against the AMP-BP algorithm from [DSMM18] in two different settings. We set $n = 1000$ and work with two values of $d \in \{5, 500\}$. The case $d = 5$ simulates the low-dimensional case where asymptotically $d/n \to 0$, while $d = 500$ represents the high-dimensional case where $d/n \to c$ for a constant c. For the AMP-BP algorithm, we choose two values for the number of iterations, $t \in \{5, 20\}$.

Tables 5.1 and 5.2 show the results for $\zeta = 1$ with varying values of γ. We observe that the classifier obtained after training Architecture 2 almost always outperforms AMP-BP for both low-dimensional and high-dimensional cases.

Table 5.1: Accuracy metrics for Architecture 2 and AMP-BP for $\zeta = 1$ and $d = 5$.

Graph signal	5-local Bayes Optimal	AMP, $t = 5$	AMP, $t = 20$
0.3	0.870	0.753	0.858
0.4	0.954	0.816	0.890
0.5	0.988	0.819	0.916
0.6	0.996	0.892	0.952

Table 5.2: Accuracy metrics for Architecture 2 and AMP-BP for $\zeta = 1$ and $d = 500$.

Graph signal	5-local Bayes Optimal	AMP, $t = 5$	AMP, $t = 20$
0.3	0.916	0.554	0.848
0.4	0.995	0.558	0.877
0.5	0.998	0.626	0.920
0.6	0.998	0.657	0.940

Tables 5.3 and 5.4 show the results for $\zeta = 0.2$, i.e., for a weaker feature signal in the data. Here, we observe that although Architecture 2 outperforms AMP-BP in the low-dimensional regime, it exhibits worse performance than AMP-BP in the high-dimensional regime.

It is important to note, however, that this is an apples-to-oranges comparison because AMP-BP is not a local algorithm, i.e., the whole graph is visible to the algorithm and all

Table 5.3: Accuracy metrics for Architecture 2 and AMP-BP for $\zeta = 0.2$ and $d = 5$.

Graph signal	5-local Bayes Optimal	AMP, $t = 5$	AMP, $t = 20$
0.3	0.554	0.520	0.579
0.4	0.587	0.528	0.584
0.5	0.823	0.604	0.756
0.6	0.997	0.637	0.880

Table 5.4: Accuracy metrics for Architecture 2 and AMP-BP for $\zeta = 0.2$ and $d = 500$.

Graph signal	5-local Bayes Optimal	AMP, $t = 5$	AMP, $t = 20$
0.3	0.584	0.502	0.543
0.4	0.706	0.525	0.694
0.5	0.762	0.568	0.804
0.6	0.788	0.604	0.878

nodes contribute to the classification of every other node. This is not true for Architecture 2, where only the nodes in the ℓ-hop neighbourhood contribute to this decision. Our notion of optimality is among the class of local algorithms. Furthermore, we observe that AMP-BP with 5 iterations does not converge and obtains a much lower accuracy compared to 20 iterations.

Chapter 6

Subsequent Relevant Work

Since the inception of the work presented in this thesis, there has been good progress in the field of understanding graph neural networks and graph data using statistically grounded techniques and tools. The most recent work in this regard is [WBF24], which provides a rigorous theoretical analysis of the performance of *corrected* graph convolutions (corrected by removing the principal eigenvector to avoid oversmoothing). It provides theoretical guarantees for both exact and partial linear classification, which was missing in all previous works, including the work in this thesis.

In particular, this new set of results gives a spectral analysis for k rounds of corrected graph convolutions for any arbitrary k and shows that each round of convolution can reduce the misclassification error exponentially for partial classification. Additionally, it is demonstrated that the separability threshold for exact classification can be improved exponentially as a function of the number of corrected convolutions.

6.1 Corrected Graph Convolutions

Let $\mathbf{A}$ be the adjacency matrix of the given graph, and $\mathbf{D}$ be the degree matrix. Vanilla graph convolutions are represented using matrices such as $\mathbf{D}^{-1}\mathbf{A}$ or $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ [KW17]. Suppose the graph is $d-$regular, meaning that each node has exactly d neighbours. In this case, both graph convolutions reduce to $\frac{1}{d}A$. The top eigenvector of $\mathbf{A}$ is $\mathbf{1}$ with eigenvalue d, where $\mathbf{1}$ is the vector of all ones. This means that $\lim_{k\to\infty}\frac{1}{d^k}\mathbf{A}^k = \frac{1}{n}\mathbf{1}\mathbf{1}^\top$, which implies that multiplying our feature vector, $\mathbf{x}$, by $\frac{1}{d}\mathbf{A}$ many times is essentially equivalent to projecting our data onto the all-ones vector, $\mathbf{x} \mapsto \frac{1}{n}\langle\mathbf{x},\mathbf{1}\rangle\mathbf{1}$. This means that all feature values will converge to the same point. Therefore, we should expect, as verified by most real-world and synthetic experiments, that many rounds of the convolution $\mathbf{x} \mapsto \frac{1}{d}\mathbf{A}\mathbf{x}$ will lead to a large learning error. However, if we instead apply the corrected graph convolution $\tilde{\mathbf{A}} := \frac{1}{d}\mathbf{A} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ instead, then the convergent behaviour of $\mathbf{x} \mapsto \tilde{\mathbf{A}}^k$ would be equivalent to projecting $\mathbf{x}$ onto the *second* eigenvector of $\mathbf{A}$. This eigenvector is known to capture information about sparse cuts in graphs [Che70, AM85, Alo86], and so for certain problems, like binary classification, we may expect this eigenvector to capture a larger amount of information about our signal.

All the results in this chapter are stated in terms of the following corrected convolution matrices:

$$\hat{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} - \frac{1}{\mathbf{1}^\top\mathbf{D}\mathbf{1}}\mathbf{D}^{1/2}\mathbf{1}\mathbf{1}^\top\mathbf{D}^{1/2} \qquad \text{and} \qquad \tilde{\mathbf{A}} = \frac{1}{d}\mathbf{A} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top, \tag{6.1}$$

where d is the empirical average degree in $\mathbf{A}$, which is equal to $2|E|/n$ and $|E|$ is the number of edges in the graph. Note that $\hat{\mathbf{A}}$ is derived from the *normalized* adjacency matrix, while $\tilde{\mathbf{A}}$ is (up to a scalar multiple) its *unnormalized* counterpart. Briefly, in our results, we demonstrate that when the graph is sufficiently dense and of reasonably good quality, the corrected graph convolutions exponentially improve both partial and exact classification guarantees. Depending on the density and quality of the given graph, improvement becomes saturated after $\mathcal{O}(\log n)$ convolutions in our partial and exact classification results. However, in comparison to a similar analysis in [WCWJ23] for vanilla graph convolutions (without correction), we show that classification accuracy does not become worse as the number of convolutions increases.

6.2 Data Model

A specialized version of the CSBM (see Section 2.2) is used to obtain these results. In particular, the model is defined by parameters $n, m \in \mathbb{N}$, $p, q \in [0,1]$, $\boldsymbol{\mu}, \boldsymbol{\nu}, \in \mathbb{R}^m$ and $\sigma \in \mathbb{R}^+$, with $\mathbf{P} = \{\mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbf{I}_m), \mathcal{N}(\boldsymbol{\nu}, \sigma^2\mathbf{I}_m\}$ and $\mathbf{Q} = \begin{pmatrix} p & q \\ q & p \end{pmatrix}$. The vertices of the feature-decorated graph $G(V,E) = (\mathbf{A}, \mathbf{X})$ are thus partitioned into two classes, denoted by S and T, each of size $n/2$. The goal is to recover this partition. For each $i \in V$, let $\mathbf{g}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_m)$ be an iid Gaussian noise vector. For $i \in S$, the i^{th} row of $\mathbf{X}$ is given by $\boldsymbol{\mu} + \mathbf{g}_i$, while for each $j \in T$, the j^{th} row of $\mathbf{X}$ is given by $\boldsymbol{\nu} + \mathbf{g}_j$. For ease of notation, assume in this chapter that $p > q$, however, note that the results can be easily generalized to the other symmetric case $p < q$. In the following, $\gamma(p,q) = \frac{|p-q|}{p+q}$.

6.3 Partial Classification

The partial classification or "detection" problem aims to correctly classify $1-o(1)$ fraction of vertices with probability $1-o(1)$. The following result illustrates the detection threshold.

Theorem 11. *Suppose* $(\mathbf{A}, \mathbf{X}) \sim \mathrm{CSBM}(n, p, q, \{\boldsymbol{\mu}, \boldsymbol{\nu}\}, \sigma)$ *where* $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^m$ *are means of the Gaussian mixture, and* p, q *satisfy* $\gamma(p,q) := \frac{p-q}{p+q} \geq \Omega\left(\sqrt{\frac{1}{np}}\right)$ *and* $p \geq \Omega\left(\frac{\log^2 n}{n}\right)$*. Then, there exists a linear classifier such that after* k *rounds of convolution with* $\tilde{\mathbf{A}}$*, with probability at least* $1 - \frac{1}{2}\exp(-\Omega\left(\frac{n\|\boldsymbol{\mu}-\boldsymbol{\nu}\|^2}{\sigma^2}\right))$ *it misclassifies at most*

$$O\left(\frac{1}{\gamma^2 p} + \Big(\frac{C}{\gamma\sqrt{np}}\Big)^{2k} \frac{\sigma^2}{\|\boldsymbol{\mu}-\boldsymbol{\nu}\|^2} n \log n\right)$$

vertices, where C *is an absolute constant. Furthermore, if* $\gamma \geq \Omega(\sqrt{\frac{\log n}{np}})$ *then with probability at least* $1-\frac{1}{2}\exp(-\Omega\left(\frac{n\|\boldsymbol{\mu}-\boldsymbol{\nu}\|^2}{\sigma^2}\right))$*, the same linear classifier after* k *rounds of convolution with* $\hat{\mathbf{A}}$ *will misclassify at most*

$$O\left(\frac{\log n}{\gamma^2 p} + \Big(\frac{C\log n}{\gamma\sqrt{np}}\Big)^{2k} \frac{\sigma^2}{\|\boldsymbol{\mu}-\boldsymbol{\nu}\|^2} n \log n\right)$$

vertices.

Let us take a closer look at the error bound. First, we see that an important ratio in the bound is the term $1/(\gamma^2 np)$. This term is small if γ^2 is much larger than the inverse of the expected degree of each vertex, $(p+q)n/2$, which is at most np. The assumption that this ratio is upper bounded by a constant means that we need the signal from the graph to be sufficiently strong. Now, if we let $\rho = C/(\gamma^2 np)$ where C is a sufficiently large constant, then we see that the *fraction* of misclassified vertices is at most $\rho + \rho^k \sigma^2 \log n / \|\boldsymbol{\mu}-\boldsymbol{\nu}\|^2$. Our assumption on the parameters ensures that $\rho < 1$. Note that only the second term depends on k, and the feature's noise-to-signal ratio. This term measures the amount of error introduced by the variance in the features and exponentially decreases with k. Moreover, after about $k = \log_{1/\rho}\left(\sigma^2 \log n/(\rho\|\boldsymbol{\mu}-\boldsymbol{\nu}\|^2)\right)$ convolutions, the ρ term, which only depends on graph parameters, will dominate over the variance term, indicating that more convolutions will not improve the quality of the convolved features beyond the quality of the signal from the graph. If $\|\boldsymbol{\mu}-\boldsymbol{\nu}\|/\sigma$ is constant, we will always reach the optimal error bound of $O(\rho)$ when $k = O(\log\log n)$. Moreover, if $\gamma = \Omega(1)$, as was assumed in

[BFJ21b], we have $1/\rho \geq \Omega(np)$. This means that even when $\sigma / \|\boldsymbol{\mu} - \boldsymbol{\nu}\| \approx \sqrt{n/\log n}$, we will reach optimality in a constant number of convolutions with a high probability if the graph is moderately dense. For example, if $p = 1/\sqrt{n}$, then we only need 3 convolutions and if $p = \Omega(1)$, then we only need 2. On the other hand, if γ is on the order of $\Theta(1/\sqrt{np})$, then in the worst case, we may need $\log n$ convolutions to reach the optimal bound.

6.4 Exact Classification

The exact classification objective aims to recover S and T with probability $1 - o(1)$.

Theorem 12. *Suppose* $(\mathbf{A}, \mathbf{X}) \sim \mathrm{CSBM}(n, p, q, \{\boldsymbol{\mu}, \boldsymbol{\nu}\}, \sigma)$ *where* $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^m$ *are means of the Gaussian mixture, and* p, q *satisfy* $\gamma(p,q) \geq \Omega\left(k\sqrt{\frac{\log n}{np}}\right)$ *and* $p \geq \frac{\log^3 n}{n}$. *Then after* $k = O(\log n)$ *rounds of graph convolution with* $\tilde{\mathbf{A}}$, *the data is linearly separable with probability* $1 - n^{-\Omega(1)}$ *if*

$$\frac{\|\boldsymbol{\mu} - \boldsymbol{\nu}\|}{\sigma} \geq \Omega\left(\max\left(\sqrt{\frac{\log n}{n}}, \left(\frac{C}{\gamma\sqrt{np}}\right)^k \sqrt{\log n}\right)\right)$$

where C *is an absolute constant.*

Theorem 12 provides the minimum signal-to-noise ratio required for exact classification as a function of n, p, q, k. Similar to the partial classification result, this function has a term that decreases exponentially with k and a term that is independent of k. The rate of decrease of the dependent term is proportional to $1/(\gamma\sqrt{np})$, or $\sqrt{\rho}$. We see once again that with more convolutions, the requirement on the feature signal-to-noise ratio for exact classification becomes exponentially weaker. Moreover, because of the assumption that $\gamma \geq \Omega(k\sqrt{\log n/(np)})$, as long as $\|\boldsymbol{\mu} - \boldsymbol{\nu}\| \geq \Omega(\sigma\sqrt{\log n/n})$, the data becomes linearly separable after $k = O(\log n / \log\log n)$ convolutions. Observe that the larger γ is, the fewer convolutions we need to obtain the optimal bound. In particular, if $\gamma = \Omega(1)$ and $p = \Omega(1)$ then one convolution already gives the optimal bound, and if $p = 1/\sqrt{n}$, then two convolutions are enough.

6.5 Discussion on Assumptions

For the complete proofs of Theorems 11 and 12, the reader is referred to [WBF24, Theorem 4.1 and 4.2]. Note that both the theorems require a lower bound of $\gamma \geq \omega(1/\sqrt{np})$ to ensure that the signal from the graph is strong enough so that a convolution does not destroy the signal from the feature data. Furthermore, implicit in the probability bound of Theorems 11 and 12 is the assumption that the signal-to-noise ratio, $\|\boldsymbol{\mu} - \boldsymbol{\nu}\| / \sigma$, is at least $\omega(1/\sqrt{n})$ so the feature signal is sufficiently strong. The lower-bound assumption on p is to ensure the concentration of degrees and the adjacency matrix towards its expectation. In Theorem 12, it is also assumed that $k = O(\log n)$, mainly for technical reasons of the proof. Finally, note that the case $p > q$ corresponds to a homophilous graph, and the case $p < q$ corresponds to a heterophilous graph (see [LHL$^+$21b, MLST22] for more). For binary classification, it has been shown [BFJ23a] that one can assume $p > q$ without loss of generality and make corresponding adjustments in the classifier. As such, we assume that $p > q$.

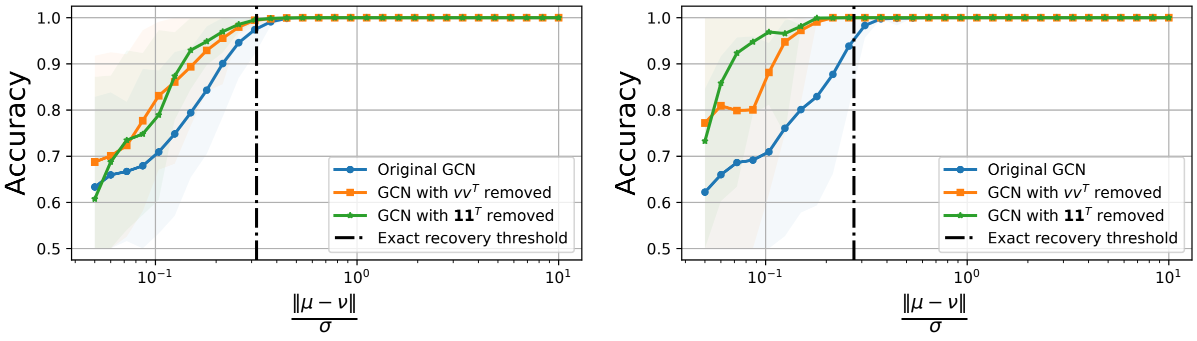
6.6 Experiments

This section demonstrates the above results empirically. For synthetic data, Theorems 11 and 12 are simulated for linear binary classification. For real data, it is shown that removing the principal component of the adjacency matrix exhibits positive effects on multi-class node classification problems as well.

6.6.1 Synthetic Data

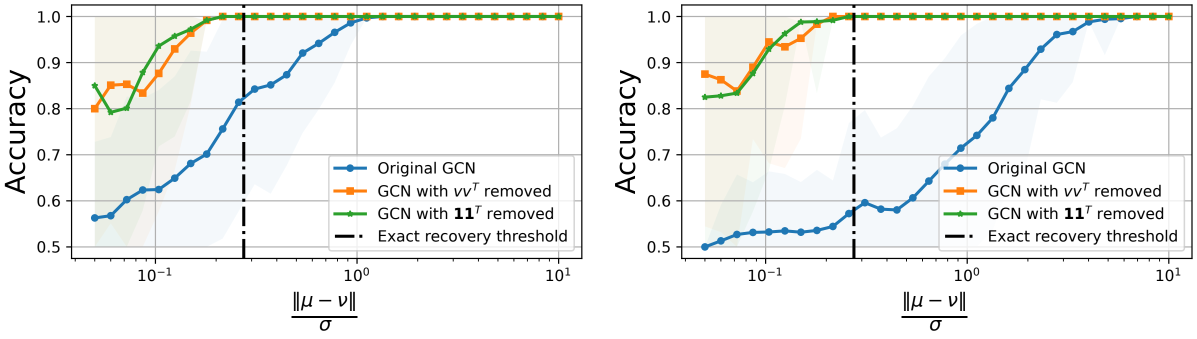
For synthetic data from the CSBM, we observe the benefits of removing the principal component of the adjacency matrix before performing convolutions for both variants of convolution described in (6.1). The experiments are performed for $n = 2000$ nodes with 20 features for each node, sampled from a Gaussian mixture. The intra-edge probability is fixed to $p = O(\log^3 n/n)$. Linear classification is performed to demonstrate the results, training a one-layer GCN both with and without the corrected convolutions.

Plots are provided for two different settings: (1) Fix $\gamma = |p - q|/(p + q) = 2/3$ and vary signal-to-noise ratio of the node features, $\|\boldsymbol{\mu} - \boldsymbol{\nu}\|/\sigma$, for a different number of convolutions. We observe in Fig. 6.1 that as the number of convolutions increases, the original GCN [KW17] (in blue) starts performing poorly, while the corrected versions (in orange and green) retain the accuracy for lower signal-to-noise ratio; (2) Fix $\|\boldsymbol{\mu} - \boldsymbol{\nu}\|/\sigma = 1$ and vary the graph relative signal strength, γ, for different number of convolutions. We observe the same trends in this setting, as depicted in Fig. 6.2. The vertical lines represent the threshold for exact classification from Theorem 12.



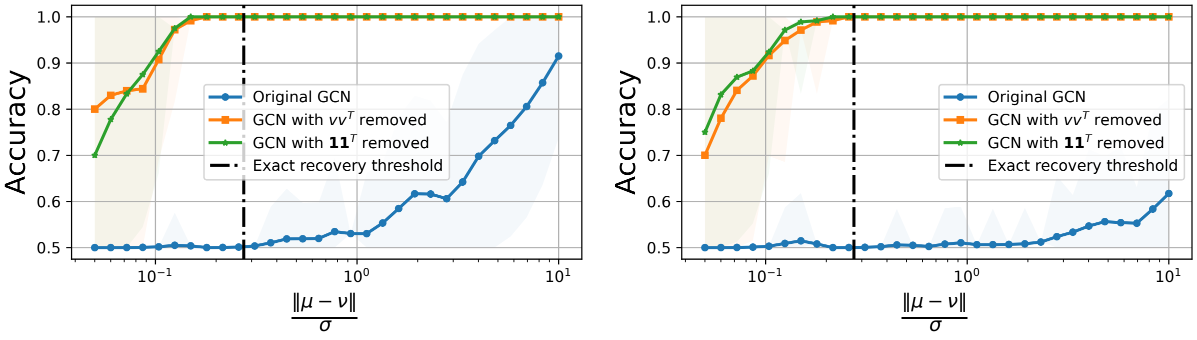
(a) 1 convolution.

(b) 2 convolutions.



(c) 4 convolutions.

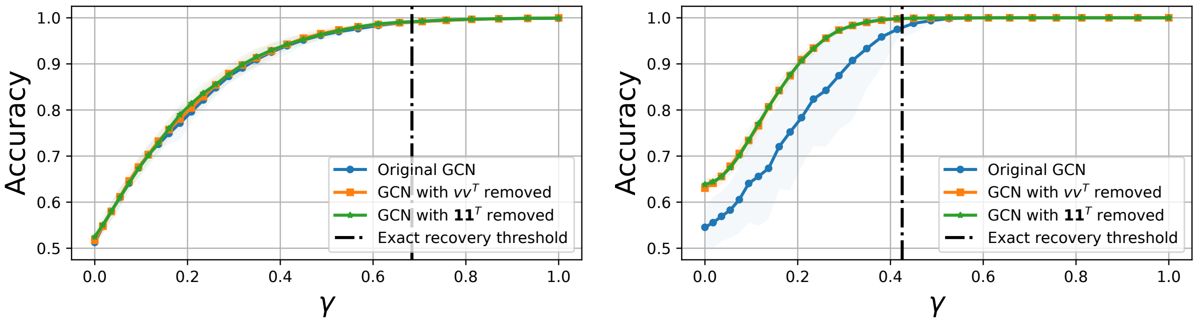
(d) 8 convolutions.



(e) 12 convolutions.

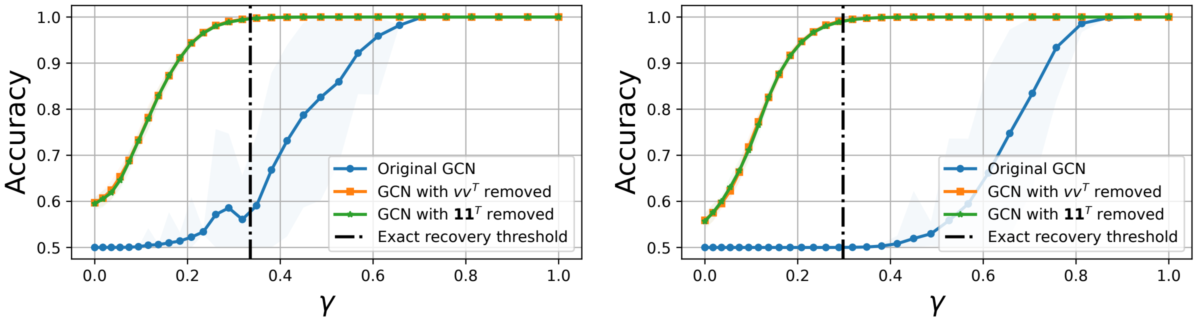
(f) 16 convolutions.

Figure 6.1: Accuracy plot (averaged over 50 trials) against the signal-to-noise ratio of the features (ratio of the distance between the means to the standard deviation) for a given number of convolutions. Here, $v = \mathbf{D}^{1/2}\mathbf{1}$ and the "GCN with $vv^\top$ removed" refers to convolution with the corrected, normalized adjacency matrix. "GCN with $\mathbf{1}\mathbf{1}^\top$ removed" is the corrected, unnormalized matrix.



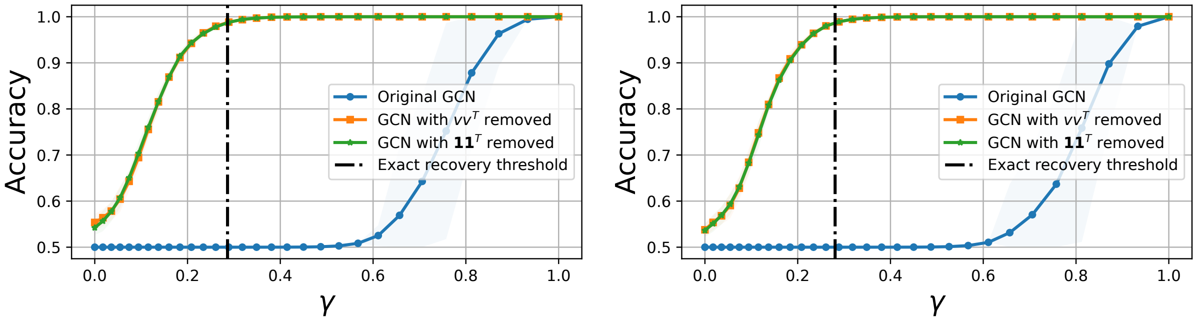
(a) 1 convolution.

(b) 2 convolutions.



(c) 4 convolutions.

(d) 8 convolutions.

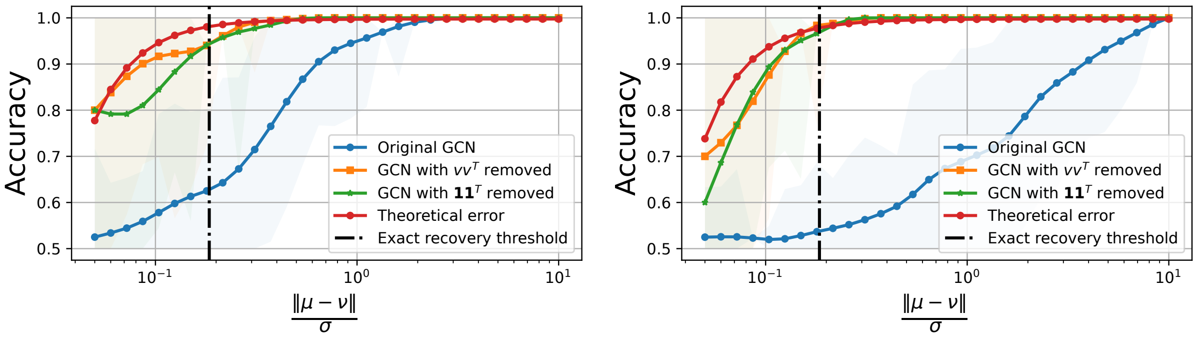


(e) 12 convolutions.

(f) 16 convolutions.

Figure 6.2: Accuracy plot (averaged over 50 trials) against graph relative signal strength ($\gamma = |p - q|/(p + q)$) for various values of the number of convolutions.

The partial classification result is demonstrated in Fig. 6.3, which plots the theoretical value of the error from Theorem 11 along with the simulated values. We observe that for suitable values of the constants in Theorem 11, the simulated accuracy (fraction of correctly classified nodes) is roughly the same as the theoretical bound.



(a) 4 convolutions.

(b) 8 convolutions.

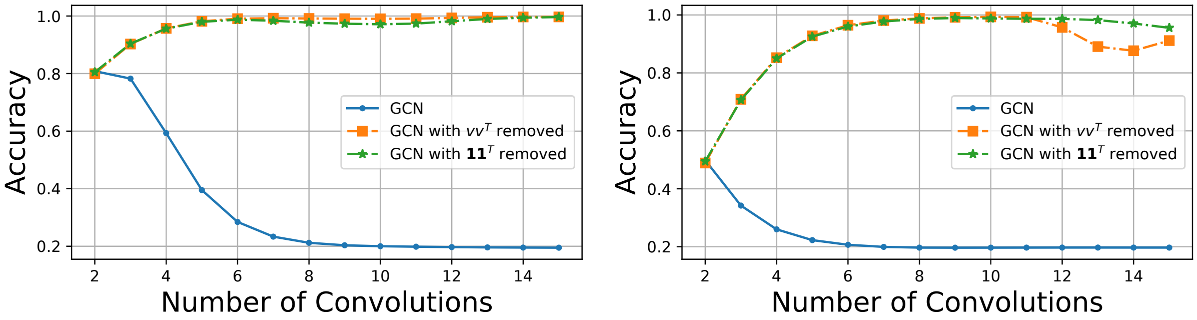
Figure 6.3: The theoretical error as derived from Theorem 11 is shown in red.

Next, let us depict a comparison between the corrected vs uncorrected convolutions via both linear and non-linear models for multi-class classification against the number of convolutions. For the linear model (Section 6.6.1), we look at five 1-vs-all classifiers followed by a softmax to predict the class label, while the non-linear method (Section 6.6.1) follows a typical two-layer MLP-based architecture. In both cases, we observe that the corrected convolutions do not deteriorate in performance as the number of convolutions increases.

6.6.2 Real Data

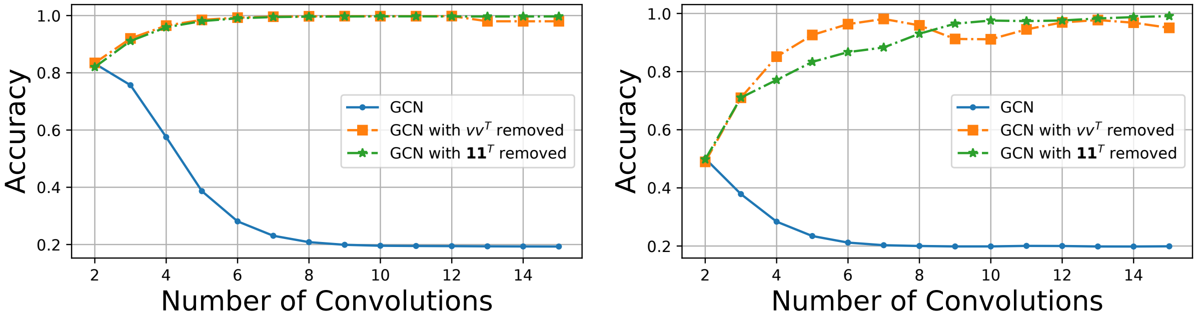
Similar to synthetic data, the results for the corrected and the original GCN are compared on the following real graph benchmark datasets: *CORA*, *CiteSeer*, and *Pubmed* citation networks [SNB+08]. It is observed in Fig. 6.5 that correcting the convolution as in Eq. (6.1) by removing the principal component performs better as the number of layers increases, even for a multi-class setting, where we perform one convolution at each layer.

Note that in Fig. 6.5, we see that overall the accuracy of every learning method decreases as the number of convolutions increases but the corrected convolutions converge to an accuracy much higher than that of the uncorrected convolution. This is attributed to the fact that for multi-class classification, the important information about class memberships is captured by the top C eigenvectors (except the first one) where C is the number of classes [LGT14]. Since the limiting behaviour of many rounds of convolutions is akin to projecting the features onto the second eigenvector, we only expect this to capture partial information about the multi-class structure.



(a) Linear method, $q = 0.02$, $\sigma = 8$.

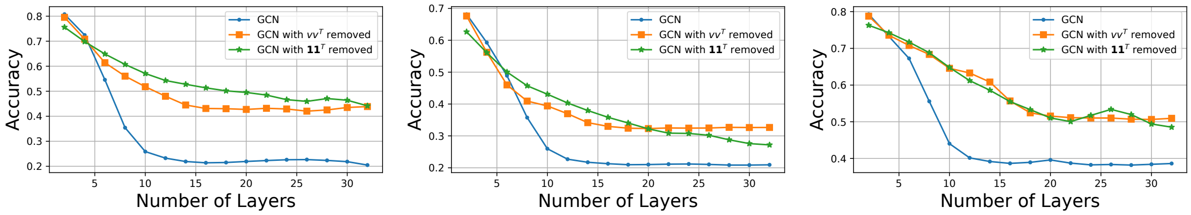
(b) Linear method, $q = 0.04$, $\sigma = 8$.



(c) Nonlinear method, $q = 0.02$, $\sigma = 8$.

(d) Nonlinear method, $q = 0.04$, $\sigma = 8$.

Figure 6.4: Accuracy plot (averaged over 50 trials) on CSBM data with 5 balanced classes, 500 nodes per class and orthogonal means, with fixed $p = 0.1$.



(a) Cora.

(b) CiteSeer.

(c) PubMed.

Figure 6.5: Accuracy plots against the number of layers for real datasets.

Chapter 7

Conclusion and Future Work

This thesis attempts to build a statistically grounded methodology for understanding, evaluating, and developing learning methods for graph machine learning problems. It provides a rigorous theoretical framework which enables a broader, yet fundamental understanding of learning on graphs, using well-studied random graph models coupled with node attribute information. There are several limitations of this work, for example, it specifically deals with the node classification problem. However, I hope future research in this field can extend the ideas presented in this thesis to understand a wider variety of learning problems in wider, more challenging regimes.

The first part of this thesis uses the binary contextual stochastic block model to show that graph convolution can transform data which is not linearly separable into data which is linearly separable. It also showed empirically that graph convolution can be disadvantageous if the intra-class edge probability is close to the inter-class edge probability. Furthermore, a classifier trained on the convolved data can generalize to out-of-distribution data with different intra and inter-class edge probabilities. The study of binary models is then extended to understand the fundamental classification thresholds for graph convolutions when placed beyond the first layer in a multi-layer network. The XOR-CSBM data model is used to provide theoretical guarantees for the performance of graph convolutions in different regimes of the signal in the data. Through experiments on both synthetic and real-world data, it is shown that the number of convolutions is a more significant factor for determining the performance of a network, rather than the number of layers in the network. Furthermore, placing graph convolutions in any combination achieves mutually similar performance enhancements for the same number of them. Multiple graph convolutions are advantageous only when the underlying graph is relatively sparse. Intuitively, this is because, in a dense graph, a single convolution can gather information from a large number of nodes, while in a sparser graph, more convolutions are needed to gather information from a larger number of nodes (Chapter 3).

In the second part, the impact of graph attention on edges and its implications for node classification are studied. It is shown that graph attention may not be very useful in a "hard" regime where the node features are noisy by proving that a single-layer graph attention convolution is limited when it comes to distinguishing intra-class from inter-class edges (Chapter 4). Given the empirical successes of graph attention and its many variants, a promising future work is to study the power of multi-layer graph attention convolutions for distinguishing intra-class and inter-class edges. Variants of graph attention networks have been successfully used in tasks other than node classification, such as link prediction and graph classification. These tasks are typically solved by architectures that add a final aggregation layer which combines node representations generated from graph attention convolution. It is an interesting future direction to develop a good understanding of the benefits and limitations of the graph attention mechanism for these tasks.

Our analysis in these two parts is limited to a regime where the graph is relatively dense, i.e., $p, q = \Omega(\log^2 n/n)$. To fully understand the limitations of graph convolutions, a study of their effects in the case of very sparse graphs (up to $p, q = O(1/n)$) is warranted. This problem is hard because degree concentration does not provide a useful bound in this extremely sparse regime. Without degree concentration, it is difficult to argue about the nature of the output of the neural networks with overwhelming probability. Consequently, it is difficult to show a classification result. However, this level of sparsity comes with a benefit: the local neighbourhoods of every node are tree-like. This fact is crucial to the work in part three of this thesis, where it is made precise and used to design the optimal message-passing architecture for sparse graphs.

Part three presents a comprehensive theoretical characterization of the Bayes optimal node classification architecture for sparse feature-decorated graphs and shows that it can be realized through a GNN using the message-passing framework. Utilizing the well-established and well-studied statistical model CSBM, its performance is interpreted in terms of the SNR in the data, along with a validation of the findings through empirical analysis of synthetic data (Chapter 5). The following limitations are identified as prospects for future work: (1) Neighbourhoods only up to distance $\ell = c \log n$ are considered for a small enough c. Extending ℓ to the graph's diameter (known to be $O(\log n)$ with high probability) by removing the restriction on c poses challenges due to the presence of cycles, which add a lot of correlation. (2) More insights can be provided through experiments on real data to benchmark the architecture in cases where we have a significant gap between the theoretical assumptions (sparse and locally tree-like graph) and the real-world data.

References

[Abb18] E. Abbe. Community Detection and Stochastic Block Models: Recent Developments. *Journal of Machine Learning Research*, 18:1–86, 2018.

[ABH15] E. Abbe, A. S. Bandeira, and G. Hall. Exact recovery in the stochastic block model. *IEEE Transactions on Information Theory*, 62(1):471–487, 2015.

[AFT$^+$18] A. Athreya, D. E. Fishkind, M. Tang, C. E. Priebe, Y. Park, J. T. Vogelstein, K. Levin, V. Lyzinski, Y. Qin, and D. L. Sussman. Statistical inference on random dot product graphs: a survey. *Journal of Machine Learning Research*, 18(226):1–92, 2018.

[Alo86] Noga Alon. Eigenvalues and Expanders. *Combinatorica*, 6:83–96, 1986.

[AM85] Noga Alon and Vitali Milman. ℓ_1-Isoperimetric Inequalities for Graphs, and Superconcentrators. *Journal of Combinatorial Theory, Series B*, 38(1):73–88, 1985.

[AS04] N. Alon and J. H. Spencer. *The Probabilistic Method.* John Wiley & Sons, 2004.

[AS15] E. Abbe and C. Sandon. Community Detection in General Stochastic Block models: Fundamental Limits and Efficient Algorithms for Recovery. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 670–688, 2015.

[AS18] E. Abbe and C. Sandon. Proof of the achievability conjectures for the general stochastic block model. *Communications on Pure and Applied Mathematics*, 71(7):1334–1406, 2018.

[AT07] R J Adler and J E Taylor. Gaussian Inequalities. In *Random Fields and Geometry*, chapter 2, pages 49–64. Springer New York, New York, NY, 2007.

[BAY22] S. Brody, U. Alon, and E. Yahav. How attentive are graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2022.

[BBCV21] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[BBK08] V. Blondel, S. Boyd, and H. Kimura. Graph implementations for nonsmooth convex programs, Recent Advances in Learning and Control (a tribute to M. Vidyasagar). *Lecture Notes in Control and Information Sciences, Springer*, pages 95–110, 2008.

[BC05] Andrew D Barbour and Louis Hsiao Yun Chen. *An introduction to Stein's method*, volume 4. World Scientific, 2005.

[BCB15] D. Bahdanau, K. H. Cho, , and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*, 2015.

[BDGC$^+$22] Cristian Bodnar, Francesco Di Giovanni, Benjamin Chamberlain, Pietro Lio, and Michael Bronstein. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnns. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:18527–18541, 2022.

[BFJ21a] A. Baranwal, K. Fountoulakis, and A. Jagannath. Graph convolution for semi-supervised classification: Improved linear separability and out-of-distribution generalization. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139, pages 684–693, 2021.

[BFJ21b] Aseem Baranwal, Kimon Fountoulakis, and Aukosh Jagannath. Graph Convolution for Semi-Supervised Classification: Improved Linear Separability and Out-of-Distribution Generalization. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 684–693. PMLR, 18–24 Jul 2021.

[BFJ23a] Aseem Baranwal, Kimon Fountoulakis, and Aukosh Jagannath. Effects of Graph Convolutions in Multi-layer Networks. In *The Eleventh International Conference on Learning Representations*, 2023.

[BFJ23b] Aseem Baranwal, Kimon Fountoulakis, and Aukosh Jagannath. Optimality of Message-Passing Architectures for Sparse Graphs. In *Advances in Neural Information Processing Systems*, volume 36, pages 40320–40341, 2023.

[BKGB$^+$20] Victor Bapst, Thomas Keck, A Grabska-Barwińska, Craig Donner, Ekin Dogus Cubuk, Samuel S Schoenholz, Annette Obika, Alexander WR Nelson, Trevor Back, Demis Hassabis, et al. Unveiling the Predictive Power of Static Structure in Glassy Systems. *Nature Physics*, 16(4):448–454, 2020.

[BL11] Lars Backstrom and Jure Leskovec. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 635–644, 2011.

[BL17] Xavier Bresson and Thomas Laurent. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*, 2017.

[BLM15] C. Bordenave, M. Lelarge, and L. Massoulié. Non-backtracking spectrum of random graphs: community detection and non-regular ramanujan graphs. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 1347–1357. IEEE, 2015.

[BMNN16] J. Banks, C. Moore, J. Neeman, and P. Netrapalli. Information-theoretic thresholds for community detection in sparse networks. In *Conference on Learning Theory*, pages 383–416. PMLR, 2016.

[Bor16] Charles Bordenave. Lecture notes on random graphs and probabilistic combinatorial optimization, 2016.

[BPL$^+$16] P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, and K. Kavukcuoglu. Interaction Networks for Learning about Objects, Relations and Physics. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[BRH$^+$21] Muhammet Balcilar, Guillaume Renton, Pierre Héroux, Benoit Gaüzère, Sébastien Adam, and Paul Honeine. Analyzing the Expressive Power of Graph Neural Networks in a Spectral Perspective. In *International Conference on Learning Representations*, 2021.

[BVR17] N. Binkiewicz, J. T. Vogelstein, and K. Rohe. Covariate-assisted spectral clustering. *Biometrika*, 104:361–377, 2017.

[CCH$^+$22] Eli Chien, Wei-Cheng Chang, Cho-Jui Hsieh, Hsiang-Fu Yu, Jiong Zhang, Olgica Milenkovic, and Inderjit S Dhillon. Node Feature Extraction by Self-Supervised Multi-scale Neighborhood Prediction. In *International Conference on Learning Representations*, 2022.

[Che70] Jeff Cheeger. A Lower Bound for the Smallest Eigenvalue of the Laplacian. *Problems in Analysis*, pages 195–199, 1970.

[Che75] Louis H. Y. Chen. Poisson Approximation for Dependent Trials. *The Annals of Probability*, 3(3):534 – 545, 1975.

[CL01] Fan Chung and Linyuan Lu. The diameter of sparse random graphs. *Advances in Applied Mathematics*, 26(4):257–279, 2001.

[CLB19] Z. Chen, L. Li, and J. Bruna. Supervised Community Detection with Line Graph Neural Networks. In *International Conference on Learning Representations (ICLR)*, 2019.

[CPLM20] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Joint adaptive feature smoothing and topology extraction via generalized pagerank gnns. *arXiv preprint arXiv:2006.07988*, 2020.

[CVCB19] Zhengdao Chen, Soledad Villar, Lei Chen, and Joan Bruna. On the equivalence between graph isomorphism testing and function approximation with GNNs. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[CWH$^+$20] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and Deep Graph Convolutional Networks. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1725–1735. PMLR, 13–18 Jul 2020.

[CZY11] Hong Cheng, Yang Zhou, and Jeffrey Xu Yu. Clustering Large Attributed Graphs: A Balance between Structural and Attribute Similarities. *ACM Transactions on Knowledge Discovery from Data*, 12, 2011.

[DAM15] Y. Deshpande, E. Abbe, and A. Montanari. Asymptotic mutual information for the two-groups stochastic block model. *ArXiv*, 2015. arXiv:1507.08685.

[DBV16] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems (NeurIPS)*, page 3844–3852, 2016.

[DKLX$^+$18] Jan Dreier, Philipp Kuinke, Ba Le Xuan, et al. Local Structure Theorems for Erdos–Rényi Graphs and Their Algorithmic Applications. *SOFSEM 2018: Theory and Practice of Computer Science LNCS 10706*, page 125, 2018.

[DKMZ11] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84(6):066106, 2011.

[DSMM18] Yash Deshpande, Subhabrata Sen, Andrea Montanari, and Elchanan Mossel. Contextual Stochastic Block Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[DV12] T. A. Dang and E. Viennet. Community Detection based on Structural and Attribute Similarities. In *The Sixth International Conference on Digital Society (ICDS)*, 2012.

[FL19] M. Fey and J. E. Lenssen. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[FLY$^+$23] Kimon Fountoulakis, Amit Levi, Shenghao Yang, Aseem Baranwal, and Aukosh Jagannath. Graph Attention Retrospective. *Journal of Machine Learning Research*, 24(246):1–52, 2023.

[GB13] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.0 beta. http://cvxr.com/cvx, 2013.

[GBG19] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *International Conference on Learning Representations*, 2019.

[GFRS13] Stephan Günnemann, Ines Färber, Sebastian Raubach, and Thomas Seidl. Spectral Subspace Clustering for Graphs with Feature Vectors. In *IEEE 13th International Conference on Data Mining*, 2013.

[GJJ20] V. Garg, S. Jegelka, and T. Jaakkola. Generalization and Representational Limits of Graph Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 119, pages 3419–3430, 2020.

[GRS22] Julia Gaudio, Miklos Z Racz, and Anirudh Sridhar. Exact Community Recovery in Correlated Stochastic Block Models. *arXiv preprint arXiv:2203.15736*, 2022.

[GSGG23] Lukas Gosch, Daniel Sturm, Simon Geisler, and Stephan Günnemann. Revisiting Robustness in Graph Machine Learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[GSR$^+$17] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural Message Passing for Quantum Chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[GVB12] Jerome Gilbert, Ernest Valveny, and Horst Bunke. Graph Embedding in Vector Spaces by Node Attribute Statistics. *Pattern Recognition*, 45(9):3072–3083, 2012.

[Ham20] William L Hamilton. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning*, 14(3):1–159, 2020.

[Han14] R. Van Handel. Probability in high dimension. Technical report, Princeton University NJ, 2014.

[HFZ$^+$20a] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec. Open graph benchmark: datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, 2020.

[HFZ$^+$20b] 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. *arXiv preprint arXiv:2005.00687*, 2020.

[HLL83] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

[HYL17] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035, 2017.

[HZC$^+$19] Y. Hou, J. Zhang, J. Cheng, K. Ma, R. T. B. Ma, H. Chen, and M.-C. Yang. Measuring and improving the use of graph information in graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

[HZC$^+$20] Yifan Hou, Jian Zhang, James Cheng, Kaili Ma, Richard T. B. Ma, Hongzhi Chen, and Ming-Chang Yang. Measuring and Improving the Use of Graph Information in Graph Neural Networks. In *International Conference on Learning Representations*, 2020.

[Jeg22] S. Jegelka. Theory of graph neural networks: Representation and learning. In *arXiv:2204.07697*, 2022.

[JLL$^+$19] Di Jin, Ziyang Liu, Weihao Li, Dongxiao He, and Weixiong Zhang. Graph Convolutional Networks Meet Markov Random Fields: Semi-Supervised Community Detection in Attribute Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 3(1):152–159, 2019.

[JMLV22] Adrián Javaloy, Pablo Sanchez Martin, Amit Levi, and Isabel Valera. Learnable Graph Convolutional Attention Networks. In *Has it Trained Yet? NeurIPS 2022 Workshop*, 2022.

[KBV21a] N. Keriven, A. Bietti, and S. Vaiter. On the universality of graph neural networks on large random graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[KBV21b] Nicolas Keriven, Alberto Bietti, and Samuel Vaiter. On the Universality of Graph Neural Networks on Large Random Graphs. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 6960–6971. Curran Associates, Inc., 2021.

[Ker22] Nicolas Keriven. Not Too Little, Not Too Much: A theoretical Analysis of Graph (Over)Smoothing. In *Advances in Neural Information Processing Systems*, 2022.

[KTA19] B. Knyazev, G. W. Taylor, and M. Amer. Understanding attention and generalization in graph neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4202–4212, 2019.

[KUK17] I. M. Kloumann, J. Ugander, and J. Kleinberg. Block models and personalized PageRank. *Proceedings of the National Academy of Sciences*, 114(1):33–38, 2017.

[KW17] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*, 2017.

[LCM19] P. Li, I. (Eli) Chien, and O. Milenkovic. Optimizing Generalized PageRank Methods for Seed-Expansion Community Detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 11705–11716, 2019.

[LGT14] James R. Lee, Shayan Oveis Gharan, and Luca Trevisan. Multiway Spectral Partitioning and Higher-Order Cheeger Inequalities. *J. ACM*, 61(6), dec 2014.

[LHL$^+$21a] Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 20887–20902, 2021.

[LHL$^+$21b] Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Is Heterophily A Real Nightmare For Graph Neural Networks To Do Node Classification? *arXiv preprint arXiv:2109.05641*, 2021.

[LHL$^+$22] Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Revisiting heterophily for graph neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 1362–1375, 2022.

[LHW18] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In *Thirty-Second AAAI conference on artificial intelligence*, 2018.

[LJZ$^+$22] Songtao Liu, Shixiong Jing, Tong Zhao, Zengfeng Huang, and Dinghao Wu. Enhancing Multi-hop Connectivity for Graph Convolutional Networks. In *First Workshop on Pre-training: Perspectives, Pitfalls, and Paths Forward at ICML 2022*, 2022.

[LMBB18] Ron Levie, Federico Monti, Xavier Bresson, and Michael M Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2018.

[Lou20a] A. Loukas. How hard is to distinguish graphs with graph neural networks? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[Lou20b] A. Loukas. What Graph Neural Networks Cannot Learn: Depth vs Width. In *International Conference on Learning Representations (ICLR)*, 2020.

[LPW$^+$17] Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The Expressive Power of Neural Networks: A View from the Width. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[LRK$^+$19] B. J. Lee, R. A. Rossi, S. Kim, K. N. Ahmed, and E. Koh. Attention models in graphs: A survey. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2019.

[LS20] Chen Lu and Subhabrata Sen. Contextual Stochastic Block Model: Sharp Thresholds and Contiguity. *ArXiv*, 2020. arXiv:2011.09841.

[LZBT16] Y. Li, R. Zemel, M. Brockschmidt, and D. Tarlow. Gated graph sequence neural networks. In *International Conference on Learning Representations (ICLR)*, 2016.

[Mas14] Laurent Massoulié. Community Detection Thresholds and the Weak Ramanujan Property. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, page 694–703, 2014.

[MBHSL19] Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and Equivariant Graph Networks. In *International Conference on Learning Representations*, 2019.

[MBM$^+$17] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M. Bronstein. Geometric Deep Learning on Graphs and Manifolds Using Mixture Model CNNs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[MDR19] Nikhil Mehta, Lawrence Carin Duke, and Piyush Rai. Stochastic Blockmodels meet Graph Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 4466–4474, 2019.

[MLLK22a] S. Maskey, R. Levie, Y. Lee, and G. Kutyniok. Generalization analysis of message passing neural networks on large random graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[MLLK22b] Sohir Maskey, Ron Levie, Yunseok Lee, and Gitta Kutyniok. Generalization Analysis of Message Passing Neural Networks on Large Random Graphs. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[MLST22] Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. Is Homophily a Necessity for Graph Neural Networks? In *International Conference on Learning Representations*, 2022.

[MNS15a] E. Mossel, J. Neeman, and A. Sly. Consistency thresholds for the planted bisection model. In *Proceedings of the forty-seventh annual ACM Symposium on Theory of computing*, pages 69–75, 2015.

[MNS15b] Elchanan Mossel, Joe Neeman, and Allan Sly. Reconstruction and estimation in the planted partition model. *Probability Theory and Related Fields*, 162:431–461, 2015.

[MNS18] E. Mossel, J. Neeman, and A. Sly. A proof of the block model threshold conjecture. *Combinatorica*, 38(3):665–708, 2018.

[Moo17] C. Moore. The Computer Science and Physics of Community Detection: Landscapes, Phase Transitions, and Hardness. *Bulletin of The European Association for Theoretical Computer Science*, 1(121), 2017.

[MS16] A. Montanari and S. Sen. Semidefinite programs on sparse random graphs and their application to community detection. In *Proceedings of the forty-eighth annual ACM Symposium on Theory of Computing*, pages 814–827, 2016.

[MSRR19] Ryan Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Relational Pooling for Graph Representations. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4663–4673. PMLR, 09–15 Jun 2019.

[OS20] Kenta Oono and Taiji Suzuki. Graph Neural Networks Exponentially Lose Expressive Power for Node Classification. In *International Conference on Learning Representations*, 2020.

[Owe80] D. B. Owen. A table of normal integrals. *Communications in Statistics-Simulation and Computation*, 9(4):389–419, 1980.

[PBHL20] O. Puny, H. Ben-Hamu, and Y. Lipman. Global attention improves graph networks generalization. In *arXiv:2006.07846*, 2020.

[Ram21] Kavita Ramanan. CRM-PIMS Summer School 2021: Background Material For Mini Course on Asymptotics of Interacting Stochastic Processes on Sparse Graphs, 2021.

[RH15] P. Rigollet and J.-C. Hütter. High dimensional statistics. *Lecture notes for course 18S997*, 813:814, 2015.

[RHXH20] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. In *International Conference on Learning Representations*, 2020.

[SGT⁺09] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1), 2009.

[SNB⁺08] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93, 2008.

[Ste72] Charles Stein. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proc. Sixth Berkeley Symp. Math. Stat. Prob.*, pages 583–602, 1972.

[VCC⁺18a] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations (ICLR)*, 2018.

[VCC⁺18b] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *International Conference on Learning Representations*, 2018.

[Ver18] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*, volume 47. Cambridge University Press, 2018.

[VSP⁺17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, page 6000–6010, 2017.

[WBF24] Robert Wang, Aseem Baranwal, and Kimon Fountoulakis. Analysis of corrected graph convolutions. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[WCWJ23] Xinyi Wu, Zhengdao Chen, William Wei Wang, and Ali Jadbabaie. A Non-Asymptotic Analysis of Oversmoothing in Graph Neural Networks. In *The Eleventh International Conference on Learning Representations*, 2023.

[WDC$^+$19] Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I Weidele, Claudio Bellei, Tom Robinson, and Charles E Leiserson. Anti-Money Laundering in Bitcoin: Experimenting with Graph Convolutional Networks for Financial Forensics. *arXiv preprint arXiv:1908.02591*, 2019.

[WYJ$^+$22] Rongzhe Wei, Haoteng Yin, Junteng Jia, Austin R. Benson, and Pan Li. Understanding Non-linearity in Graph Neural Networks from the Bayesian-Inference Perspective. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[WZY$^+$19] M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, J. Zhou, C. Ma, L. Yu, Y. Gai, T. Xiao, T. He, G. Karypis, J. Li, and Z. Zhang. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*, 2019.

[WZYW22] Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf. Towards Distribution Shift of Node-Level Prediction on Graphs: An Invariance Perspective. In *International Conference on Learning Representations*, 2022.

[XHLJ19] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? *In International Conference on Learning Representations (ICLR)*, 2019.

[XLT$^+$18] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation Learning on Graphs with Jumping Knowledge Networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5453–5462. PMLR, 10–15 Jul 2018.

[XZL$^+$21] Keyulu Xu, Mozhi Zhang, Jingling Li, Simon S. Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. How Neural Networks Extrapolate: From Feedforward to Graph Neural Networks. In *International Conference on Learning Representations*, 2021.

[YHC$^+$18] 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*, pages 974–983, 2018.

[YHS$^+$21] 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, 2021.

[YHS$^+$22] 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. In *IEEE International Conference on Data Mining (ICDM)*, pages 1287–1292. IEEE, 2022.

[YML13] J. Yang, J. McAuley, and J. Leskovec. Community Detection in Networks with Node Attributes. In *2013 IEEE 13th International Conference on Data Mining*, pages 1151–1156, 2013.

[ZYZ$^+$20] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Generalizing graph neural networks beyond homophily. *arXiv preprint arXiv:2006.11468*, 2020.

[ZZY$^+$17] Si Zhang, Dawei Zhou, Mehmet Yigit Yildirim, Scott Alcorn, Jingrui He, Hasan Davulcu, and Hanghang Tong. Hidden: Hierarchical Dense Subgraph Detection with Application to Financial Fraud Detection. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 570–578. SIAM, 2017.