Quantifying the Optimization and Generalization Advantages of Graph Neural Networks Over Multilayer Perceptrons

Wei Huang[1] **Yuan Cao**[2] **Haonan Wang**[3] **Xin Cao**[4] **Taiji Suzuki**[5,1]
[1]RIKEN AIP [2]The University of Hong Kong [3]National University of Singapore
[4]The University of New South Wales [5]University of Tokyo

Abstract

Graph neural networks (GNNs) have demonstrated remarkable capabilities in learning from graph-structured data, often outperforming traditional Multilayer Perceptrons (MLPs) in numerous graph-based tasks. Although existing works have demonstrated the benefits of graph convolution through Laplacian smoothing, expressivity or separability, there remains a lack of quantitative analysis comparing GNNs and MLPs from an optimization and generalization perspective. This study aims to address this gap by examining the role of graph convolution through feature learning theory. Using a signal-noise data model, we conduct a comparative analysis of the optimization and generalization between two-layer graph convolutional networks (GCNs) and their MLP counterparts. Our approach tracks the trajectory of signal learning and noise memorization in GNNs, characterizing their post-training generalization. We reveal that GNNs significantly prioritize signal learning, thus enhancing the regime of low test error over MLPs by D^{q-2} times, where D denotes a node's expected degree and q is the power of ReLU activation function with $q > 2$. This finding highlights a substantial and quantitative discrepancy between GNNs and MLPs in terms of optimization and generalization, a conclusion further supported by our empirical simulations on both synthetic and real-world datasets.

1 Introduction

Graph neural networks (GNNs) have recently demonstrated remarkable capability in learning graph representations, yielding superior results across various downstream tasks, such as node classifications [32, 50, 21], graph classifications [55, 20, 36, 60] and link predictions [33, 61, 35], etc. Compared to multilayer perceptron (MLPs), GNNs enhance representation learning with an added message passing operation [64]. Take graph convoluational network (GCN) [32] as an example, it aggregates a node's attributes with those of its neighbors through a *graph convolution* operation. This operation, which leverages the structural information (adjacency matrix) of graph data, forms the core distinction between GNNs and MLPs.

There is substantial evidence showing that GCNs consistently outperform MLPs empirically [32, 46, 64, 48, 12, 17, 41]. This success has led to numerous theoretical studies aimed at understanding the underlying reasons for the effectiveness of graph convolutions. One prominent explanation is that graph convolutions leverage Laplacian smoothing to filter out noise from input features, as highlighted by several works [39, 9, 44, 45]. Another line of research investigates the expressivity of graph neural networks in distinguishing complex graph structures [18, 40, 55]. Recent studies have also explored the theoretical role of graph convolutions in enhancing separability. For instance, [5] considered a setting of linear classification of data generated from a contextual stochastic block model [15], showing that graph convolution extends the regime where data is linearly separable by a factor of approximately $1/\sqrt{D}$ compared to MLPs, with D denoting a node's expected degree. Building on this, [6] further examined multi-layer graph nerual networks and demonstrated improved non-linear separability with the incorporation of graph convolutions.

Despite the valuable insights provided by the existing literature, there has been limited theoretical research that offers a *quantitative* understanding of the opti-

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

mization and generalization properties of GNNs compared to their MLP counterparts in a unified framework. To address this gap, we aim to answer the following key question from a theoretical perspective:

What role does graph convolution play during gradient descent training, and to what extent do GCNs exhibit better generalization compared to MLPs?

To address this critical question, we conduct a feature learning analysis [8, 3] for graph neural networks to establish a unified theoretical framework that analyzes both the convergence and generalization properties of GCNs, enabling a *quantitative comparison* with MLPs. Specifically, we introduce a data generation model, termed, that combines a signal-noise model [2, 8] for input feature creation with a stochastic block model [1] for graph construction. This setting serves as a representative case study for our analysis. Our theoretical investigation focuses on the optimization trajectory and post-training generalization of two-layer GCNs trained using gradient descent, and compares these results with the established findings for two-layer MLPs [8]. While both GCNs and MLPs are shown to achieve near-zero training error, our analysis reveals a distinct quantitative advantage for GCNs in terms of generalization on test data. The key contributions of our study are as follows:

- **Global Convergence Guarantees:** We provide global convergence guarantees for graph neural networks trained on data generated by the SNM-SBM model. By characterizing signal learning and noise memorization in feature learning, we show that, despite the inherent nonconvexity of the optimization landscape, GCNs can achieve zero training error within a polynomial number of iterations.

- **Test Error Bounds for Overfitted GNNs:** We derive theoretical test error bounds for overfitted GNN models trained using gradient descent. Under specific conditions on the signal-to-noise ratio, we demonstrate that GCNs can achieve small (near-zero) test error, even when the model is over-fitted to the noisy data.

- **Quantitative Generalization Comparison:** We provide a quantitative comparison of the generalization performance between GCNs and MLPs. Our results identify a regime where GCNs achieve nearly zero test error, while MLPs exhibit a significantly higher test error. This finding is further validated through empirical experiments.

2 Related Work

Role of Graph Convolution in GNNs. Existing studies [12, 41, 63, 23, 51] have shown that graph convolutions significantly enhance the performance of traditional classification methods, such as MLPs. Motivated by these benefits, various graph network architectures and methods have been proposed to further harness the power of graph convolutions [53, 19, 54]. From a theoretical standpoint, [57] highlight the superiority of GNNs for extrapolation problems in comparison with MLPs, based on the graph neural tangent kernel (GNTK) [28, 16, 26, 47]. [26] use a similar approach to examine the role of graph convolution in deep GNNs, revealing that excessive graph layers can degrade optimization and generalization, thus supporting the well-known over-smoothing problem in deep GNNs [39]. In addition, [58] attribute the major performance gains of GNNs to their inherent generalization capability. [10] primarily examine the expressivity of Graph-Augmented MLPs. Similarly, [24] propose two smoothness metrics to quantify the quality of information extracted from graph data and introduce a novel attention-based framework to optimize GNN performance. Besides, [56] investigate the impact of skip connections, increased depth, and favorable label distributions on the training dynamics of GNNs. In contrast to these existing theoretical results, our work moves beyond GNTK analysis and differs from common approaches like Laplacian smoothing, expressivity, or separability studies. Instead, we focus on developing a unified framework to quantitatively compare GNNs and MLPs through a comprehensive convergence and generalization analysis.

Feature Learning in Deep Learning. This work builds upon a growing body of research on how neural networks learn features. [2] formulate a theory illustrating that when data possess a "multi-view" feature, ensembles of independently trained neural networks can demonstrably improve test accuracy. Further, [3] show that adversarial training can eliminate specific small dense mixtures from the hidden layers, thereby refining the learned weights. Additionally, [4] identify that the initial gradient update introduces a rank-1 'spike', waligning the first-layer weights with the linear component of the teacher model's features. The seminal work by [8] investigates the benign overfitting phenomenon in two-layer convolutional neural networks (CNNs) and reveals that under certain signal-to-noise ratio conditions, gradient descent can drive a two-layer CNN to achieve near-zero test loss through effective feature learning. Related works [59, 65, 52, 14, 66, 11, 43, 29, 34, 13, 37, 25, 22, 30, 7, 27] have similarly highlighted the role of feature learning in neural networks during gradient descent training, forming a critical area of research that our study continues to explore. While [38, 62] apply feature learning theory to analyze GNNs, they do not quantify the differences in optimization and generalization between GNNs and MLPs. Our work addresses this gap by providing a *quantitative comparison*, offering new insights into the distinct behaviors of these architectures.

3 Problem Setup

3.1 Notations

We use lower bold-faced letters for vectors, upper bold-faced letters for matrices, and non-bold-faced letters for scalars. For a vector $\mathbf{v}$, its ℓ_2-norm is denoted as $\|\mathbf{v}\|_2$. For a matrix $\mathbf{A}$, we use $\|\mathbf{A}\|_2$ to denote its spectral norm and $\|\mathbf{A}\|_F$ for its Frobenius norm. We employ standard asymptotic notations such as $O(\cdot)$, $o(\cdot)$, $\Omega(\cdot)$, and $\Theta(\cdot)$ to describe the limiting behavior. We use $\widetilde{O}(\cdot)$, $\widetilde{\Omega}(\cdot)$, and $\widetilde{\Theta}(\cdot)$ to hide logarithmic factors in these notations respectively. Moreover, we denote $a_n = \mathrm{poly}(b_n)$ if $a_n = O((b_n)^p)$ for some positive constant p and $a_n = \mathrm{polylog}(b_n)$ if $a_n = \mathrm{poly}(\log(b_n))$. Lastly, sequences of integers are denoted as $[m] = \{1, 2, \dots, m\}$.

3.2 Data Model

We adopt a combined *signal-noise model* (SNM) for feature generation and a *stochastic block model* for graph structure generation as a case study.

Signal-noise Model Specifically, let the feature matrix be denoted as $\mathbf{X} \in \mathbb{R}^{n\times 2d}$, where n represents the number of samples and $2d$ is the feature dimensionality. Each feature associated with a data point is generated from a SNM, conditional on the Rademacher random variable $y \in \{-1, 1\}$, and a latent vector $\boldsymbol{\mu} \in \mathbb{R}^d$:

$$\mathbf{x} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}] = [y\boldsymbol{\mu}, \boldsymbol{\xi}], \tag{1}$$

where $\mathbf{x}^{(1)}, \mathbf{x}^{(2)} \in \mathbb{R}^d$, and $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \sigma_p^2 \cdot (\mathbf{I} - \|\boldsymbol{\mu}\|_2^{-2} \cdot \boldsymbol{\mu}\boldsymbol{\mu}^\top))$ with σ_p^2 as the variance. The term $\mathbf{I} - \|\boldsymbol{\mu}\|_2^{-2} \cdot \boldsymbol{\mu}\boldsymbol{\mu}^\top$ ensures that the noise vector $\boldsymbol{\xi}$ is orthogonal to the signal vector $\boldsymbol{\mu}$.

We make the following remarks on the data model.

- **Feature Composition:** The data model simulates a setting where the input features consist of both signal and noise components. In particular, the term $y\boldsymbol{\mu}$ represents task (label)-relevant features, while $\boldsymbol{\xi}$ captures task-irrelevant features. Recent works [2, 8, 65, 49] have explored similar signal-noise models to investigate the feature learning process of neural networks, including studies focused on graph neural networks [38, 62].

- **Real-world Relevance:** The SNM reflects the input feature structure of real-world graph datasets used in node classification tasks. For example, citation network datasets such as Cora, Citeseer, and Pubmed [32] typically use a bag-of-words representation for node features. Conceptually, these words can be divided into two categories: task-relevant and task-irrelevant. Words like "algorithm" or "neural network" are task-relevant for the field of computer science, whereas more generic words like "study" or "approach" are task-irrelevant.

Stochstic Block Model We employ a SBM to generate the graph structure, with intra-class edge probability p and inter-class edge probability s. Specifically, each entry in the adjacency matrix $\mathbf{A} = (a_{ij})_{n\times n}$ follows a Bernoulli distribution: $a_{ij} \sim \mathrm{Ber}(p)$ when $y_i = y_j$, and $a_{ij} \sim \mathrm{Ber}(s)$ when $y_i \neq y_j$. The use of p and s are explicitly modeled, allows us to explicitly model different graph structures and analyze the impact of varying connectivity patterns.

When $p \gg s$, the graph structure exhibits *homophily*, meaning that nodes are more likely to connect with others that share the same label, resulting in clusters of similarly labeled nodes. Conversely, when $s \gg p$, the graph reflects a *heterophily*, where nodes are more likely to connect to those with different labels. This flexibility makes SBM a powerful tool for simulating diverse graph topologies, and it is widely used in related studies on GNNs [5, 42, 6, 41, 31].

We represent the combination of the SNM and SBM as $\mathrm{SNM} - \mathrm{SBM}(n, p, s, \boldsymbol{\mu}, \sigma_p, d)$. This combined model captures both feature and structural variations, providing a unified framework for studying GNNs. Note that when $p = s = 0$, the graph structure disappears, and $\mathrm{SNM} - \mathrm{SBM}$ reduces to the standard SNM.

3.3 Graph Neural Network Model

Graph neural network (GNNs) integrate both graph structure and node features to learn meaningful representations for nodes. Consider a two-layer GCN model, denoted as f, where the first layer performs a graph convolution operation, and the second layer parameters are fixed to either $+1$ or -1. The output of the GCN is given by:

$$f(\mathbf{W}, \mathbf{A}, \mathbf{x}) = F_{+1}(\mathbf{W}_{+1}, \mathbf{A}, \mathbf{x}) - F_{-1}(\mathbf{W}_{-1}, \mathbf{A}, \mathbf{x}),$$

where $F_{+1}(\mathbf{W}_{+1}, \mathbf{A}, \mathbf{x})$ and $F_{-1}(\mathbf{W}_{-1}, \mathbf{A}, \mathbf{x})$ are defined as:

$$F_j(\mathbf{W}_j, \mathbf{A}, \mathbf{x}) = \frac{1}{m} \sum_{r=1}^{m} \left[\sigma(\mathbf{w}_{j,r}^\top \tilde{\mathbf{x}}^{(1)}) + \sigma(\mathbf{w}_{j,r}^\top \tilde{\mathbf{x}}^{(2)}) \right]. \tag{2}$$

where $j \in \{+1, -1\}$, and $\mathbf{W}_{\pm 1}$ refer to the first layer weights associated with the second-layer fixed parameters. Besides, $\sigma(\cdot)$ is a polynomial ReLU activation function defined as $\sigma(z) = \max\{0, z\}^q$ for some $q > 2$, and m is the width of hidden layer. The notation $\tilde{\mathbf{X}} \triangleq [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \cdots, \tilde{\mathbf{x}}_n]^\top = \tilde{\mathbf{D}}^{-1}\tilde{\mathbf{A}}\mathbf{X} \in \mathbb{R}^{n \times 2d}$ represents the node features after the graph convolution.

Specifically, the adjacency matrix with self-loops is defined as $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_n$, where $\mathbf{A}$ is the original adjacency matrix and $\mathbf{I}_n$ is the identity matrix of size n. The diagonal degree matrix $\tilde{\mathbf{D}}$ records the degree of each node, with entries given by $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. For simplicity we denote $D_i \triangleq \tilde{D}_{ii}$.

Remark 3.1. The use of a polynomial ReLU activation function ensures a significant gap between signal learning and noise memorization. This type of activation function has been widely adopted in related works [2, 3, 8, 65, 34] to facilitate the theoretical study of neural network training dynamics.

Moreover, the symbol $\mathbf{W}$ collectively denotes the first layer's weights, and $\mathbf{w}_{j,r} \in \mathbb{R}^d$ refers to the weight of the first layer, in which r corresponds to hidden neuron index and $j \in \{+1, -1\}$ refer to the fixed value of second layer. The first-layer weights are initialized by sampling from a Gaussian distribution, i.e., $\mathbf{w}_{j,r} \sim \mathcal{N}(\mathbf{0}, \sigma_0^2 \cdot \mathbf{I}_{d \times d})$ for all $r \in [m]$ and $j \in \{-1, 1\}$, where σ_0 controls the strength of the initial weights.

Given the training data $\mathcal{S} \triangleq \{\mathbf{x}_i, y_i\}_{i=1}^n$ and adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ drawn from $\mathrm{SNM-SBM}(n, p, s, \boldsymbol{\mu}, \sigma_p, d)$, we aim to learn the parameter $\mathbf{W}$ by minimizing the empirical cross-entropy loss function:

$$L_{\mathcal{S}}^{\mathrm{GCN}}(\mathbf{W}) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i \cdot f(\mathbf{W}, \mathbf{A}, \mathbf{x}_i)). \tag{3}$$

Here, the cross-entropy loss is defined as $\ell(y \cdot f(\mathbf{W}, \mathbf{A}, \mathbf{x})) = \log(1 + \exp(-f(\mathbf{W}, \mathbf{A}, \mathbf{x}) \cdot y))$. The gradient descent update for the first layer weight $\mathbf{W}$ in GCN can be expressed as:

$$\begin{aligned}
\mathbf{w}_{j,r}^{(t+1)} &= \mathbf{w}_{j,r}^{(t)} - \eta \cdot \nabla_{\mathbf{w}_{j,r}} L_{\mathcal{S}}^{\mathrm{GCN}}(\mathbf{W}^{(t)}) \\
&= \mathbf{w}_{j,r}^{(t)} - \frac{\eta}{nm} \sum_{i=1}^{n} \ell_i'^{(t)} \sigma'(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{y}_i \boldsymbol{\mu} \rangle) \cdot j \tilde{y}_i \boldsymbol{\mu} \\
&\quad - \frac{\eta}{nm} \sum_{i=1}^{n} \ell_i'^{(t)} \sigma'(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_i \rangle) \cdot j y_i \tilde{\boldsymbol{\xi}}_i,
\end{aligned} \tag{4}$$

where we define the loss derivative as $\ell_i' \triangleq \ell'(y_i \cdot f_i) = -\frac{\exp(-y_i \cdot f_i)}{1 + \exp(-y_i \cdot f_i)}$, the "aggregated label" $\tilde{y}_i = D_i^{-1} \sum_{k \in \mathcal{N}(i)} y_k$, and the "aggregated noise vector" $\tilde{\boldsymbol{\xi}}_i = D_i^{-1} \sum_{k \in \mathcal{N}(i)} \boldsymbol{\xi}_k$. Here $\mathcal{N}(i)$ denotes the set of neighbors of node i, $\sigma'(\cdot)$ represents the derivative of the polynomial ReLU activation function, and η is the learning rate.

To quantify the learning capabilities of GNNs compared to MLPs, we analyze the generalization ability of GNN models through the lens of population loss, which is defined based on unseen test data. After training the network on n data points, we generate a new test data point following the $\mathrm{SNM-SBM}$ distribution. Its connections to the training data points are determined using the stochastic block model, forming an updated adjacency matrix $\mathbf{A}' \in \mathbb{R}^{(n+1) \times (n+1)}$. The population loss is then calculated by taking the expectation over the randomness of the new test data, and is expressed as follows:

$$L_{\mathcal{D}}^{\mathrm{GCN}}(\mathbf{W}) = \mathbb{E}_{(\mathbf{x}, y, \mathbf{A}') \sim \mathrm{SNM-SBM}} \ell(y \cdot f(\mathbf{W}, \mathbf{A}', \mathbf{x})). \tag{5}$$

This approach for formulating the generalization error is consistent with the methodology used in [38].

3.4 Weight Decomposition for Optimization Analysis

To track the complex training dynamics described by Equation (4), we employ a weight decomposition method inspired by feature learning theory [8]. From the gradient descent rule in Equation (4), we observe that each gradient descent iterate $\mathbf{w}_{j,r}^{(t)}$ can be represented as a linear combination of its initial random weight $\mathbf{w}_{j,r}^{(0)}$, the signal vector $\boldsymbol{\mu}$, and the noise vectors $\boldsymbol{\xi}_i$[1] from the training data for $i \in [n]$. For $r \in [m]$, the weight decomposition at iteration t can be expressed:

$$\begin{aligned}
\mathbf{w}_{j,r}^{(t)} = \mathbf{w}_{j,r}^{(0)} + j\gamma_{j,r}^{(t)} \cdot \|\boldsymbol{\mu}\|_2^{-2} \cdot \boldsymbol{\mu} + \sum_{i=1}^{n} \overline{\rho}_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i \\
+ \sum_{i=1}^{n} \underline{\rho}_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i.
\end{aligned} \tag{6}$$

where $\gamma_{j,r}^{(t)}$ and $\rho_{j,r,i}^{(t)} = \{\overline{\rho}_{j,r,i}^{(t)}, \underline{\rho}_{j,r,i}^{(t)}\}$ are the coefficients of the *signal learning* and *noise memorization*, respectively. To provide a more precise characterization of the coefficients, we define $\overline{\rho}_{j,r,i}^{(t)} \triangleq \rho_{j,r,i}^{(t)} \mathbb{1}(\rho_{j,r,i}^{(t)} \geq 0)$, $\underline{\rho}_{j,r,i}^{(t)} \triangleq \rho_{j,r,i}^{(t)} \mathbb{1}(\rho_{j,r,i}^{(t)} \leq 0)$. We refer to Equation (6) as the *signal-noise decomposition* of $\mathbf{w}_{j,r}^{(t)}$. The normalization factors $\|\boldsymbol{\mu}\|_2^{-2}$ and $\|\boldsymbol{\xi}_i\|_2^{-2}$ are introduced to

[1] By referring to Equation (4), we see that the gradient descent update moves in the direction of $\tilde{\boldsymbol{\xi}}_i$, which can be further decomposed into $\boldsymbol{\xi}_i$ through $\tilde{\boldsymbol{\xi}}_i = D_i^{-1} \sum_{k \in \mathcal{N}(i)} \boldsymbol{\xi}_k$.

ensure that $\gamma_{j,r}^{(t)} \approx \langle \mathbf{w}_{j,r}^{(t)}, \boldsymbol{\mu} \rangle$, and $\rho_{j,r,i}^{(t)} \approx \langle \mathbf{w}_{j,r}^{(t)}, \boldsymbol{\xi}_i \rangle$. We use $\gamma_{j,r}^{(t)}$ to characterize the process of *signal learning* and $\rho_{j,r,i}^{(t)}$ to characterize the *noise memorization.* If certain $\gamma_{j,r}^{(t)}$ values are sufficiently large while all $|\rho_{j,r,i}^{(t)}|$ remain relatively small, this indicates that the network is primarily learning the label through signal learning. Conversely, if some $|\rho_{j,r,i}^{(t)}|$ values are large while all $\gamma_{j,r}^{(t)}$ values remain small, the network is primarily focused on noise memorization.

4 Theoretical Results

In this section, we introduce our key theoretical findings that demonstrate the optimization and generalization of GCNs through feature learning. Our analysis is based on the following assumptions:

Assumption 4.1. Suppose that (1) The dimension d is sufficiently large: $d = \tilde{\Omega}(m^{2\vee[4/(q-2)]} n^{4\vee[(2q-2)/(q-2)]})$. (2) The size of the training sample n and width of GCNs m adhere to $n, m = \Omega(\mathrm{polylog}(d))$. (3) The edge probability $p, s = \Omega(\sqrt{\log(n)/n})$ and $\Xi \triangleq \frac{p-s}{p+s}$ is a positive constant. (4) The learning rate η satisfies $\eta \leq \tilde{O}(\min\{\|\boldsymbol{\mu}\|_2^{-2}, \sigma_p^{-2} d^{-1}\})$, and the weight initialization strength follows $\sigma_0 \leq \tilde{O}(m^{-2/(q-2)} n^{-[1/(q-2)]\vee 1} \cdot \min\{(\sigma_p \sqrt{d/(n(p+s))})^{-1}, \Xi^{-1}\|\boldsymbol{\mu}\|_2^{-1}\}$.

The rationale for these assumptions is as follows: (1) The requirement for a high dimension is specifically aimed at ensuring that the learning occurs in a sufficiently over-parameterized setting when the second layer remains fixed. Similar choice can be found in [8, 34]. (2) This condition guarantees that certain statistical properties of the training data and weight initialization hold with high probability at least $1 - d^{-1}$. (3) The condition on edge probabilities p and s guarantees a sufficient concentration in node degrees and captures the level of homophily in the graph data. (4) The condition on η and σ_0 ensures that gradient descent can effectively minimize the training loss.

Furthermore, we introduce a critical quantity called the signal-to-noise ratio (SNR), which measures the relative learning speed between the signal and the noise. It is defined as $\mathrm{SNR} = \|\boldsymbol{\mu}\|_2/(\sigma_p\sqrt{d})$. To prepare for our main result, we also define an effective SNR for GNNs as $\mathrm{SNR}_G = \|\boldsymbol{\mu}\|_2/(\sigma_p\sqrt{d}) \cdot (n(p+s))^{(q-2)/(2q)}$. Given the above assumptions and definitions, we present our main result for GNNs as follows:

Theorem 4.2. *Let $T = \tilde{\Theta}(\eta^{-1} m \sigma_0^{-(q-2)} \Xi^{-q} \|\boldsymbol{\mu}\|_2^{-q} + \eta^{-1}\epsilon^{-1} m^3 \|\boldsymbol{\mu}\|_2^{-2})$. Under Assumption 4.1, if $n \cdot \mathrm{SNR}_G^q = \tilde{\Omega}(1)$, then with probability at least $1 - d^{-1}$, there exists a time $0 \leq t \leq T$ such that:*

- *The GCN learns the signal:* $\max_r \gamma_{j,r}^{(t)} = \Omega(1)$ *for* $j \in \{\pm 1\}$.
- *The GCN does not memorize the noises:* $\max_{j,r,i} |\rho_{j,r,i}^{(T)}| = \tilde{O}(\sigma_0 \sigma_p \sqrt{d/n(p+s)})$.
- *The training loss converges to ϵ, i.e., $L_{\mathcal{S}}^{\mathrm{GCN}}(\mathbf{W}^{(t)}) \leq \epsilon$.*
- *The trained GCN achieves a small test loss: $L_{\mathcal{D}}^{\mathrm{GCN}}(\mathbf{W}^{(t)}) \leq c_1\epsilon + \exp(-c_2 n^2)$, where c_1 and c_2 are positive constants.*

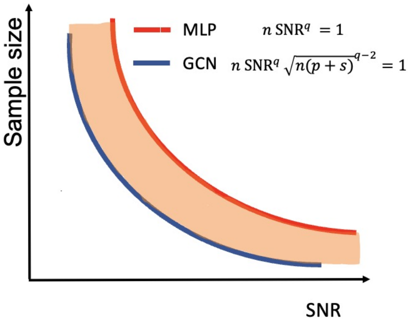


Figure 1: Illustration of test performance comparison between GNN and MLP. The region below the red curve but above the blue curve (highlighted in orange) indicates the area where GNN can generalize well, but MLP fails to generalize to the test set.

Theorem 4.2 reveals that, provided $n \cdot \mathrm{SNR}_G^q = \tilde{\Omega}(1)$, the GCN is capable of learning the signal, achieving $\max_r \gamma_{j,r}^{(t)} = \Omega(1)$. On the other hand, the noise memorization during gradient descent training is suppressed, as indicated by $\max_{j,r,i} |\rho_{j,r,i}^{(T)}| = \tilde{O}(\sigma_0\sigma_p\sqrt{d/n(p+s)})$, given that $\sigma_0\sigma_p\sqrt{d/n(p+s)} \ll 1$, according to assumption 4.1. Because the signal learned by the network is sufficiently strong and much larger than the noise memorization, the model can generalize well to the test samples. Consequently, the learned neural network achieves both low training and test losses.

We compare the result in Theorem 4.2 with the result of MLPs presented in [8] to highlight the quantitative advantage in generalization of GCNs over MLPs. According to [8], when $n \cdot \mathrm{SNR}^q = \tilde{\Omega}(1)$, MLPs can achieve a small test error; otherwise, they experience a large test error when $n \cdot \mathrm{SNR}^q = \tilde{O}(1)$. Together, these results highlight the differences in generalization, showing that GCNs have a broader regime for achieving low test errors, which is qualitatively represented by $[n(p+s)]^{(q-2)/(2q)}$. This outcome further requires a dense setting, where $n(p+s) > 1$, which is consistent with Assumption 4.1. The reason for this improved performance is that, during feature learning, graph convolution can effectively suppress noise memorization, enabling GCNs to generalize better than MLPs, especially in low SNR settings. Finally, we illustrate the quantitative difference established in this work through Figure 1. By precisely characterizing the feature learning dynamics from optimization to generalization for GNNs, we have successfully shown how GCNs gain a significant advantage over MLPs due to the benefits provided by graph convolution.

5 Proof Roadmap

In this section, we present proof sketches for GCNs using feature learning theory. We discuss the primary challenges encountered in the study of GNNs and outline the key techniques used in our proofs to address these challenges. These techniques are further elaborated in the subsequent sections, and detailed proofs can be found in the appendix. Although we adopt the feature learning framework from prior works [2, 8], our focus is fundamentally different, making the existing results not directly applicable to GCNs.

5.1 Iterative of coefficients

To analyze the feature learning process in GCNs, we introduce an iterative methodology based on the signal-noise decomposition (6) and gradient descent update rule (4). The following lemma provides a means to track the iteration of signal learning and noise memorization under graph convolution:

Lemma 5.1. *The coefficients $\gamma_{j,r}^{(t)}, \overline{\rho}_{j,r,i}^{(t)}, \underline{\rho}_{j,r,i}^{(t)}$ with the initialization value $\gamma_{j,r}^{(0)}, \overline{\rho}_{j,r,i}^{(0)}, \underline{\rho}_{j,r,i}^{(0)} = 0$ in decomposition (6) adhere to the following equations:*

$$\gamma_{j,r}^{(t+1)} = \gamma_{j,r}^{(t)} - \frac{\eta}{nm}\sum_{i=1}^{n} \ell_i'^{(t)} \sigma'(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{y}_i \boldsymbol{\mu}_i \rangle) y_i \tilde{y}_i \|\boldsymbol{\mu}\|_2^2, \tag{7}$$

$$\overline{\rho}_{j,r,i}^{(t+1)} = \overline{\rho}_{j,r,i}^{(t)} - \frac{\eta}{nm}\sum_{k\in\mathcal{N}(i)} D_k^{-1} \ell_k'^{(t)} \sigma'(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_k \rangle) \|\boldsymbol{\xi}_i\|_2^2 \mathbb{1}(y_k = j), \tag{8}$$

$$\underline{\rho}_{j,r,i}^{(t+1)} = \underline{\rho}_{j,r,i}^{(t)} + \frac{\eta}{nm}\sum_{k\in\mathcal{N}(i)} D_k^{-1} \ell_k'^{(t)} \sigma'(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_k \rangle) \|\boldsymbol{\xi}_i\|_2^2 \mathbb{1}(y_k = -j). \tag{9}$$

Lemma 5.1 simplifies the analysis of feature learning in GCNs by reducing it to the examination of the discrete dynamical system defined by Equations (7 - 9). Our proof strategy emphasizes an in-depth evaluation of the coefficient values $\gamma_{j,r}^{(t)}, \overline{\rho}_{j,r,i}^{(t)}, \underline{\rho}_{j,r,i}^{(t)}$ throughout the training process.

5.2 The importance of dense graphs in trajectory analysis

Note that graph convolution aggregates information from neighboring nodes to the central node, which often leads to a loss of *statistical concentration* for the aggregated noise vectors and labels. To mitigate this challenge, we utilize a dense graph structure by setting the edge probability $p, s = \Omega(\sqrt{\log(n)/n})$, as stated in Assumption 4.1. As a result, we show that the node degrees exhibit good concentration properties, as demonstrated by the following lemma:

Lemma 5.2. *Let $p, s = \Omega\left(\sqrt{\frac{\log(n/\delta)}{n}}\right)$ and $\delta > 0$, then with probability at least $1-\delta$, we have $n(p+s)/4 \leq D_i \leq 3n(p+s)/4$, where D_i is the degree of node i.*

Lemma 5.2 establishes the concentration of node degrees, which is crucial for the trajectory analysis of iterations for both signal learning and noise memorization.

5.3 The Role of Homophily

To preserve the sign of the graph-aggregated labels, we introduce "homophily" by setting $p > s$. This setting ensures that the convolution process effectively integrates neighborhood label information into the central node. The formal result is provided in the following lemma:

Lemma 5.3. *Suppose that $\delta > 0$, $p > s$, and $n \geq 8\frac{p+s}{(p-s)^2}\log(4/\delta)$. Then with probability at least $1-\delta$, it holds that $\frac{1}{2}\frac{p-s}{p+s}|y_i| \leq |\tilde{y}_i| \leq \frac{3}{2}\frac{p-s}{p+s}|y_i|$.*

Lemma 5.3 shows that the effective label after graph convolution retains the same sign as the original node label. It is worth noting that if we consider heterophily (where $s > p$), the same generalization results hold in our setting, except that the sign of the aggregated label will be opposite to that of the central node.

5.4 Optimization analysis

We provide a two-stage dynamics analysis to track the trajectory of the coefficients for signal learning and noise memorization, using Lemma 5.1, Lemma 5.2, and Lemma 5.3.

Stage 1. Intuitively, the initial neural network weights are small enough such that the network at initialization exhibits constant-level loss derivatives on all training data points $\ell_i'^{(0)} = \ell'[y_i \cdot f(\mathbf{W}^{(0)}, \mathbf{A}, \mathbf{x}_i)] = \Theta(1)$ for all $i \in [n]$. This is guaranteed under Assumption 4.1 on σ_0. Motivated by this observation, the dynamics of the coefficients in Equations (7 - 9) can be greatly simplified by replacing the $\ell_i'^{(t)}$ factors

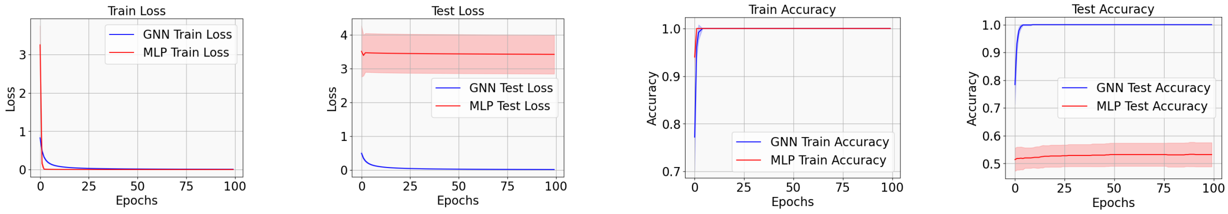


Figure 2: Training loss, test loss, training accuracy, and test accuracy for both MLP and GNN over a span of 100 training epochs. Each curve represents the average of five experimental runs, with shaded regions indicating the error bars to capture variability.

by their constant upper and lower bounds. The following lemma summarizes our main conclusion for feature learning in Stage 1:

Lemma 5.4. *Under the same conditions as Theorem 4.2, there exists* $T_1 = \tilde{O}(\eta^{-1} m \sigma_0^{2-q} \Xi^{-q} \|\boldsymbol{\mu}\|_2^{-q})$ *such that* $\max_r \gamma_{j,r}^{(T_1)} = \Omega(1)$ *for* $j \in \{\pm 1\}$, *and* $|\rho_{j,r,i}^{(t)}| = O\left(\sigma_0 \sigma_p \sqrt{d} / \sqrt{n(p+s)}\right)$ *for all* $j \in \{\pm 1\}$, $r \in [m]$, $i \in [n]$ *and* $0 \le t \le T_1$.

The proof can be found in Appendix C.1. Lemmas 5.4 leverages the period of training when the derivatives of the loss function remain at a constant order. It is important to note that graph convolution plays a significant role in differentiating the learning speeds between signal learning and noise memorization. Originally, without graph convolution, the learning speeds are primarily determined by $\|\boldsymbol{\mu}\|_2$ and $\|\boldsymbol{\xi}\|_2$ for the signal and noise, respectively [8]. In contrast, with graph convolution, the learning speeds are determined by $|\tilde{y}|\|\boldsymbol{\mu}\|_2$ and $\|\tilde{\boldsymbol{\xi}}\|_2$ respectively. Here, $|\tilde{y}|\|\boldsymbol{\mu}\|_2$ is close to $\|\boldsymbol{\mu}\|_2$, but $\|\tilde{\boldsymbol{\xi}}\|_2$ is much smaller than $\|\boldsymbol{\xi}\|_2$ (see Figure 7 for an illustration). This implies that graph convolution can slow down noise memorization, thereby allowing GNNs to focus more on signal learning.

Stage 2. Building on the results from the first stage, we move to the second stage of the training process. In this stage, the loss derivatives are no longer constant, and we show that the training error can be minimized to an arbitrarily small value. Besides, the scale differences established during the first stage of learning are preserved throughout the second stage:

Lemma 5.5. *Under the same conditions as Theorem 4.2, for any* $t \in [T_1, T]$, *it holds that* $\max_r \gamma_{j,r}^{(T_1)} \ge 2, \forall j \in \{\pm 1\}$ *and* $|\rho_{j,r,i}^{(t)}| \le \sigma_0 \sigma_p \sqrt{d/(n(p+s))}$ *for all* $j \in \{\pm 1\}$, $r \in [m]$ *and* $i \in [n]$. *Moreover, there exists a* $t \in [T_1, T]$ *such that* $L_{\mathcal{S}}^{\mathrm{GCN}}(\mathbf{W}^{(t)}) \le \epsilon$.

Lemma 5.5 presents two primary outcomes: (1) Throughout this training phase, it ensures that the noise memorization coefficients remain significantly small, while the coefficients of the signal learning reach large values. (2) It guarantees convergence for the GNN, showing that the training loss can be reduced to an arbitrarily small value.

5.5 Test error analysis

Analyzing the generalization performance of graph neural networks is a challenging task. To tackle this issue, we introduce an expectation over the distribution for a single data point. Specifically, we consider a new data point $(\mathbf{x}, y)$ drawn from the SNM-SBM distribution. The following lemma provides an upper bound on the test loss of GNNs after training:

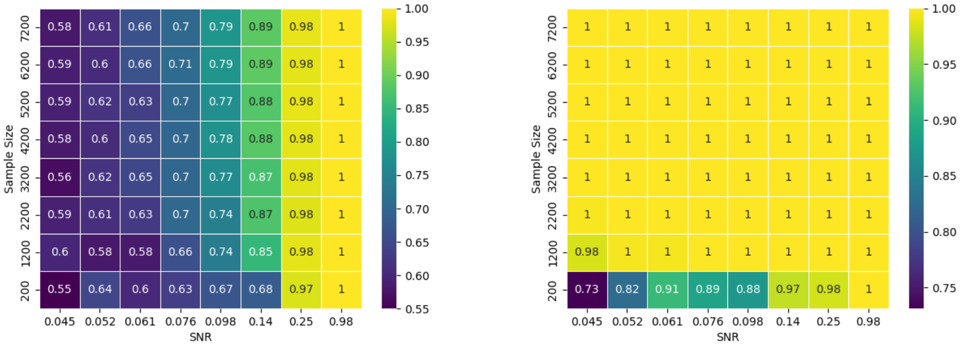
Lemma 5.6. *Let* T *be defined as in Theorem 4.2. Under the same conditions as Theorem 4.2, for any* $t \le T$ *with* $L_{\mathcal{S}}^{\mathrm{GCN}}(\mathbf{W}^{(t)}) \le 1$, *it holds that* $L_{\mathcal{D}}^{\mathrm{GCN}}(\mathbf{W}^{(t)}) \le c_1 \cdot L_{\mathcal{S}}^{\mathrm{GCN}}(\mathbf{W}^{(t)}) + \exp(-c_2 n^2)$, *where* c_1 *and* c_2 *are positive constants.*

Lemma 5.6 demonstrates that GNNs can achieve a small test error. Combined with the results stated in Lemma 5.5, this completes the proof for Theorem 4.2.

6 Experiments

In this section, we validate our theoretical findings through numerical simulations using synthetic data and modified real-world data.

Synthetic data. We generated synthetic data using the SNM-SBM model. The signal vector $\boldsymbol{\mu}$ is drawn from a standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, and the noise vector $\boldsymbol{\xi}$ is sampled from a Gaussian distribution $\mathcal{N}(\mathbf{0}, 20\mathbf{I})$. We set the training data size to $n = 50$, the input dimension to $d = 500$, the edge probability to $p = 0.5$, and $s = 0.08$. We train a two-layer MLP and GNN using Equation (2) with polynomial ReLU activation $q = 3$. The optimization is by the gradient descent method with a learning rate of $\eta = 0.03$. The primary task is node classification, aiming to predict the class labels of nodes in a graph. Figure 2 shows the training loss, test loss, training accuracy, and test accuracy for both the MLP and GNN. Our observations



(a) Performance of MLP

(b) Performance of GCN

Figure 3: Test accuracy heatmap for MLPs and GCNs after training.

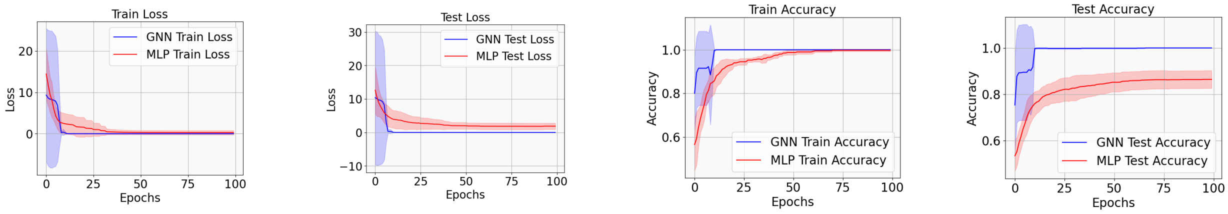


Figure 4: The verification of our theoretical result with a modified real-world data. We show the training loss, testing loss, training accuracy, and testing accuracy for both MLP and GNN over a span of 100 training epochs. Five experimental runs are conducted, with shaded areas highlighting error bars for variability.

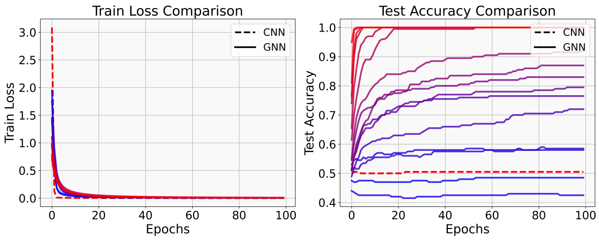


Figure 5: Training loss and test accuracy comparison between CNN (dashed lines) and GNN (solid lines) models across varying graph densities.

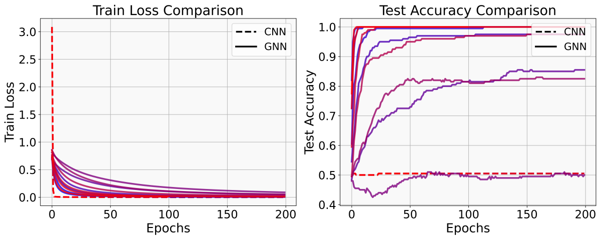


Figure 6: Training loss and test accuracy comparison between CNN (dashed lines) and GNN (solid lines) models under varying homophily levels while keeping graph density constant.

reveal that both the GNN and MLP can achieve zero training error. However, while the GNN attains nearly zero test error, the MLP fails to generalize. This simulation result validates our theoretical findings.

Graph Density We maintained a fixed ratio of $p/s = 10$, varying only the absolute values of p and s. Specifically, we considered values of p in the range $\{0.01, 0.02, 0.03, \ldots, 0.1, 0.2, \ldots, 0.7\}$. This experimental setup preserves the relative homophily while systematically adjusting the overall graph density. The results, depicted in Figure 5, utilize a color gradient from blue (sparser graphs) to red (denser graphs) to highlight changes in model performance. We observe a significant improvement in test accuracy as the graph becomes denser. Beyond a certain density threshold, the model consistently achieves near-perfect test accuracy. These findings empirically support our claim that dense graphs facilitate effective aggregation.

Graph Homophily We fixed the sum $p + s$ to a constant value, varying only the individual values of p and s. Specifically, we considered values of p within the range $\{0.1, 0.15, 0.2, \ldots, 0.6\}$. This configuration maintains a constant overall graph density while altering the level of homophily. Results are presented in Figure 6, where we employ color coding from blue (strongly homophilic) to red (strongly heterophilic). We observe that both strongly homophilic and strongly heterophilic graph structures achieve higher test accuracy, whereas graphs with intermediate homophily exhibit relatively lower performance. These findings empirically support our theoretical claims concerning the effects of homophily.

Heatmap of test accuracy We then explore a range of Signal-to-Noise Ratios (SNRs) from 0.045 to 0.98, and a variety of sample sizes, n, ranging from 200 to 7200. Based on our results, we train the neural network for 200 steps for each combination of SNR and sample size n. After training, we calculate the test accuracy for each run. The results are presented as a heatmap in Figure 3. Compared to MLPs, GCNs demonstrate a perfect accuracy score of 1 across a more extensive range in the SNR and n plane, indicating that GNNs have a broader *high test accuracy* regime with high test accuracy.

Real-world data. We conduct an experiment using the MNIST dataset. To align with the theoretical setting, we added Gaussian noise sampled from $\sigma_p^2\mathbf{I}$ to the digit images and then divide both the noise component and the digit component into two groups of patches of equal size, inspired by [8]. The details for creating this data can be found in Appendix G. We select the digits '1' and '2' from the ten MNIST digits, using a training sample size of $n = 100$, while the remaining samples are used as the test set. The graph structure was generated using a stochastic block model, with an edge probability of $p = 0.2$, and $s = 0.01$. We set the learning rate $\eta = 0.0005$ and the noise level $\sigma_p = 0.1$. Detailed results are shown in Figure 4. The results are consistent with our theoretical conclusions, reinforcing the insights derived from our analysis.

7 Conclusion and Limitation

This paper leverages feature learning theory to analyze the optimization and generalization behavior of GCNs. Specifically, we adopt a signal-noise decomposition to characterize the signal learning and noise memorization processes during the training of a two-layer GCN. We establish specific conditions under which a GNN primarily focuses on signal learning, resulting in low training and testing errors. When combined with results for MLPs, our findings quantitatively demonstrate that GCNs, by utilizing structural information, outperform MLPs in terms of generalization ability across a broader benign regime.

Limitation Our theoretical framework is limited to analyzing the role of graph convolution within a specific two-layer GCN and a particular data model. In practice, the feature learning dynamics of neural networks can be influenced by various factors, such as the depth of the GNN, the choice of activation function, the optimization algorithm, and the underlying data distribution [34, 65, 66]. Future work could extend our framework to account for the impact of these additional factors on feature learning in GCNs.

Acknowledgments

We thank the anonymous reviewers for their insightful comments to improve the paper. WH was supported by JSPS KAKENHI Grant Number 24K20848. YC was supported in part by NSFC 12301657 and Hong Kong ECS award 27308624. TS was partially supported by JSPS KAKENHI (24K02905) and JST CREST (JPMJCR2015).

References

[1] Emmanuel Abbe, Afonso S Bandeira, and Georgina Hall. Exact recovery in the stochastic block model. *IEEE Transactions on information theory*, 62(1):471–487, 2015.

[2] Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.

[3] Zeyuan Allen-Zhu and Yuanzhi Li. Feature purification: How adversarial training performs robust deep learning. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 977–988. IEEE, 2022.

[4] Jimmy Ba, Murat A Erdogdu, Taiji Suzuki, Zhichao Wang, Denny Wu, and Greg Yang. High-dimensional asymptotics of feature learning: How one gradient step improves the representation. *arXiv preprint arXiv:2205.01445*, 2022.

[5] Aseem Baranwal, Kimon Fountoulakis, and Aukosh Jagannath. Graph convolution for semi-supervised classification: Improved linear separability and out-of-distribution generalization. *arXiv preprint arXiv:2102.06966*, 2021.

[6] Aseem Baranwal, Kimon Fountoulakis, and Aukosh Jagannath. Effects of graph convolutions in multi-layer networks. In *The Eleventh International Conference on Learning Representations*, 2023.

[7] Dake Bu, Wei Huang, Andi Han, Atsushi Nitanda, Taiji Suzuki, Qingfu Zhang, and Hau-San Wong. Provably transformers harness multi-

concept word semantics for efficient in-context learning. *Advances in Neural Information Processing Systems*, 37:63342–63405, 2025.

[8] Yuan Cao, Zixiang Chen, Mikhail Belkin, and Quanquan Gu. Benign overfitting in two-layer convolutional neural networks. *arXiv preprint arXiv:2202.06526*, 2022.

[9] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3438–3445, 2020.

[10] Lei Chen, Zhengdao Chen, and Joan Bruna. On graph neural networks versus graph-augmented mlps. *arXiv preprint arXiv:2010.15116*, 2020.

[11] Yongqiang Chen, Wei Huang, Kaiwen Zhou, Yatao Bian, Bo Han, and James Cheng. Towards understanding feature learning in out-of-distribution generalization. *arXiv preprint arXiv:2304.11327*, 2023.

[12] Zhengdao Chen, Xiang Li, and Joan Bruna. Supervised community detection with line graph neural networks. *arXiv preprint arXiv:1705.08415*, 2017.

[13] Zixiang Chen, Junkai Zhang, Yiwen Kou, Xiangning Chen, Cho-Jui Hsieh, and Quanquan Gu. Why does sharpness-aware minimization generalize better than sgd? *arXiv preprint arXiv:2310.07269*, 2023.

[14] Alexandru Damian, Jason Lee, and Mahdi Soltanolkotabi. Neural networks can learn representations with gradient descent. In *Conference on Learning Theory*, pages 5413–5452. PMLR, 2022.

[15] Yash Deshpande, Andrea Montanari, Elchanan Mossel, and Subhabrata Sen. Contextual stochastic block models. *arXiv preprint arXiv:1807.09596*, 2018.

[16] Simon S Du, Kangcheng Hou, Russ R Salakhutdinov, Barnabas Poczos, Ruosong Wang, and Keyulu Xu. Graph neural tangent kernel: Fusing graph neural networks with graph kernels. *Advances in neural information processing systems*, 32, 2019.

[17] Vijay Prakash Dwivedi, Chaitanya K Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research*, 24(43):1–48, 2023.

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

[19] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018.

[20] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212*, 2017.

[21] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034, 2017.

[22] Andi Han, Wei Huang, Yuan Cao, and Difan Zou. On the feature learning in diffusion models. *arXiv preprint arXiv:2412.01021*, 2024.

[23] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648, 2020.

[24] Yifan Hou, Jian Zhang, James Cheng, Kaili Ma, Richard TB Ma, Hongzhi Chen, and Ming-Chang Yang. Measuring and improving the use of graph information in graph neural networks. *arXiv preprint arXiv:2206.13170*, 2022.

[25] Wei Huang, Andi Han, Yongqiang Chen, Yuan Cao, Zhiqiang Xu, and Taiji Suzuki. On the comparison between multi-modal and single-modal contrastive learning. *Advances in Neural Information Processing Systems*, 37:81549–81605, 2025.

[26] Wei Huang, Yayong Li, Weitao Du, Richard Yi Da Xu, Jie Yin, Ling Chen, and Miao Zhang. Towards deepening graph neural networks: A gntk-based optimization perspective. *arXiv preprint arXiv:2103.03113*, 2021.

[27] Wei Huang, Ye Shi, Zhongyi Cai, and Taiji Suzuki. Understanding convergence and generalization in federated learning through feature learning theory. In *The Twelfth International Conference on Learning Representations*, 2023.

[28] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and

generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[29] Samy Jelassi, Michael Sander, and Yuanzhi Li. Vision transformers provably learn spatial structure. *Advances in Neural Information Processing Systems*, 35:37822–37836, 2022.

[30] Jiarui Jiang, Wei Huang, Miao Zhang, Taiji Suzuki, and Liqiang Nie. Unveil benign overfitting for transformer in vision: Training dynamics, convergence, and generalization. *Advances in Neural Information Processing Systems*, 37:135464–135625, 2025.

[31] Nicolas Keriven and Samuel Vaiter. What functions can graph neural networks compute on random graphs? the role of positional encoding. *Advances in Neural Information Processing Systems*, 36, 2024.

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

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

[34] Yiwen Kou, Zixiang Chen, Yuanzhou Chen, and Quanquan Gu. Benign overfitting for two-layer relu networks. *arXiv preprint arXiv:2303.04145*, 2023.

[35] Ajay Kumar, Shashank Sheshar Singh, Kuldeep Singh, and Bhaskar Biswas. Link prediction techniques, applications, and performance: A survey. *Physica A: Statistical Mechanics and its Applications*, 553:124289, 2020.

[36] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. Self-attention graph pooling. *arXiv preprint arXiv:1904.08082*, 2019.

[37] Hongkang Li, Meng Wang, Sijia Liu, and Pin-Yu Chen. A theoretical understanding of shallow vision transformers: Learning, generalization, and sample complexity. *arXiv preprint arXiv:2302.06015*, 2023.

[38] Hongkang Li, Meng Wang, Tengfei Ma, Sijia Liu, Zaixi Zhang, and Pin-Yu Chen. What improves the generalization of graph transformer? a theoretical dive into self-attention and positional encoding. 2023.

[39] 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.

[40] Andreas Loukas. How hard is to distinguish graphs with graph neural networks? *Advances in neural information processing systems*, 33:3465–3476, 2020.

[41] Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. Is homophily a necessity for graph neural networks? *arXiv preprint arXiv:2106.06134*, 2021.

[42] Nikhil Mehta, Lawrence Carin Duke, and Piyush Rai. Stochastic blockmodels meet graph neural networks. In *International Conference on Machine Learning*, pages 4466–4474. PMLR, 2019.

[43] Xuran Meng, Yuan Cao, and Difan Zou. Per-example gradient regularization improves learning signals from noisy data. *arXiv preprint arXiv:2303.17940*, 2023.

[44] Hoang Nt and Takanori Maehara. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*, 2019.

[45] Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. *arXiv preprint arXiv:1905.10947*, 2019.

[46] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. *arXiv preprint arXiv:1907.10903*, 2019.

[47] Mahalakshmi Sabanayagam, Pascal Esser, and Debarghya Ghoshdastidar. Representation power of graph convolutions: Neural tangent kernel analysis. *arXiv preprint arXiv:2210.09809*, 2022.

[48] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

[49] Ruoqi Shen, Sebastien Bubeck, and Suriya Gunasekar. Data augmentation as feature manipulation. In *International Conference on Machine Learning*, pages 19773–19808. PMLR, 2022.

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

[51] Kun Wang, Guohao Li, Shilong Wang, Guibin Zhang, Kai Wang, Yang You, Xiaojiang Peng, Yuxuan Liang, and Yang Wang. The snowflake hypothesis: Training deep gnn with one node one receptive field. *arXiv preprint arXiv:2308.10051*, 2023.

[52] Zixin Wen and Yuanzhi Li. Toward understanding the feature learning process of self-supervised contrastive learning. In *International Conference on Machine Learning*, pages 11112–11122. PMLR, 2021.

[53] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR, 2019.

[54] 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.

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

[56] Keyulu Xu, Mozhi Zhang, Stefanie Jegelka, and Kenji Kawaguchi. Optimization of graph neural networks: Implicit acceleration by skip connections and more depth. In *International Conference on Machine Learning*, pages 11592–11602. PMLR, 2021.

[57] 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. *arXiv preprint arXiv:2009.11848*, 2020.

[58] Chenxiao Yang, Qitian Wu, Jiahua Wang, and Junchi Yan. Graph neural networks are inherently good generalizers: Insights by bridging gnns and mlps. *arXiv preprint arXiv:2212.09034*, 2022.

[59] Greg Yang and Edward J Hu. Feature learning in infinite-width neural networks. *arXiv preprint arXiv:2011.14522*, 2020.

[60] Hao Yuan and S. Ji. Structpool: Structured graph pooling via conditional random fields. In *ICLR*, 2020.

[61] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31, 2018.

[62] Shuai Zhang, Meng Wang, Pin-Yu Chen, Sijia Liu, Songtao Lu, and Miao Liu. Joint edge-model sparse learning is provably efficient for graph neural networks. *arXiv preprint arXiv:2302.02922*, 2023.

[63] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1):1–23, 2019.

[64] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

[65] Difan Zou, Yuan Cao, Yuanzhi Li, and Quanquan Gu. Understanding the generalization of adam in learning neural networks with proper regularization. *arXiv preprint arXiv:2108.11371*, 2021.

[66] Difan Zou, Yuan Cao, Yuanzhi Li, and Quanquan Gu. The benefits of mixup for feature learning. *arXiv preprint arXiv:2303.08433*, 2023.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes] Please refer to the corresponding context in Sections 3 and 4.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes] We have shown the sample complexity in Theorem 4.2.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes] We have uploaded the code as supplementary material.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes] We have stated the complete set of assumptions in Assumption 4.1.
   (b) Complete proofs of all theoretical results. [Yes] The proof sketch and complete proof are provided in Section 5 and Appendices A, B, C, and D, respectively.
   (c) Clear explanations of any assumptions. [Yes] We have provided explanations below Assumption 4.1.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a

URL). [Yes] We have uploaded the code as supplementary material and provided instructions in Section 6 and Appendices F and G.

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes] We have uploaded the code as supplementary material and provided details in Section 6 and Appendices F and G.

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes] We have included the error bars in Figures 2 and 4.

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] We have included the computing infrastructure details in Appendix H.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

   (a) Citations of the creator If your work uses existing assets. [Yes] We have provided the citation.

   (b) The license information of the assets, if applicable. [Yes] We have uploaded the code to the supplementary material.

   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable] This paper does not release new assets.

   (d) Information about consent from data providers/curators. [Not Applicable] Our study does not involve external data requiring consent.

   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable] Our study does not include any sensitive or personally identifiable information.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

   (a) The full text of instructions given to participants and screenshots. [Not Applicable] Our study does not involve any human participants.

   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable] There are no human subjects involved in our research.

   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable] No participants were recruited or compensated.

A Preliminary Lemmas

In this section, we present preliminary lemmas which form the foundation for the proofs to be detailed in the subsequent sections. The proof will be developed after the lemmas presented.

A.1 Preliminary Lemmas without Graph Convolution

In this section, we introduce necessary lemmas that will be used in the analysis without graph convolution, following the study of feature learning in CNN [8]. In particular, Lemma A.1 states that noise vectors are "almost orthogonal" to each other and Lemma A.2 indicates that random initialization results in a controllable inner product between the weights at initialization and the data vectors.

Lemma A.1. *[8] Suppose that $\delta > 0$ and $d = \Omega(\log(4n/\delta))$. Then with probability at least $1 - \delta$,*

$$\sigma_p^2 d/2 \leq \|\boldsymbol{\xi}_i\|_2^2 \leq 3\sigma_p^2 d/2,$$
$$|\langle \boldsymbol{\xi}_i, \boldsymbol{\xi}_{i'} \rangle| \leq 2\sigma_p^2 \cdot \sqrt{d \log(4n^2/\delta)},$$

for all $i, i' \in [n]$.

Lemma A.2. *[8] Suppose that $d = \Omega(\log(nm/\delta))$, $m = \Omega(\log(1/\delta))$. Then with probability at least $1 - \delta$,*

$$|\langle \mathbf{w}_{j,r}^{(0)}, \boldsymbol{\mu} \rangle| \leq \sqrt{2\log(8m/\delta)} \cdot \sigma_0 \|\boldsymbol{\mu}\|_2,$$
$$|\langle \mathbf{w}_{j,r}^{(0)}, \boldsymbol{\xi}_i \rangle| \leq 2\sqrt{\log(8mn/\delta)} \cdot \sigma_0 \sigma_p \sqrt{d},$$

for all $r \in [m]$, $j \in \{\pm 1\}$ and $i \in [n]$. Moreover,

$$\sigma_0 \|\boldsymbol{\mu}\|_2/2 \leq \max_{r \in [m]} j \cdot \langle \mathbf{w}_{j,r}^{(0)}, \boldsymbol{\mu} \rangle \leq \sqrt{2\log(8m/\delta)} \cdot \sigma_0 \|\boldsymbol{\mu}\|_2,$$
$$\sigma_0 \sigma_p \sqrt{d}/4 \leq \max_{r \in [m]} j \cdot \langle \mathbf{w}_{j,r}^{(0)}, \boldsymbol{\xi}_i \rangle \leq 2\sqrt{\log(8mn/\delta)} \cdot \sigma_0 \sigma_p \sqrt{d},$$

for all $j \in \{\pm 1\}$ and $i \in [n]$.

A.2 Preliminary Lemmas on Graph Properties

We now introduce important lemmas that are critical to our analysis. The key idea to ensure a relatively dense graph. In a sparser graph, the concentration properties of graph degree (Lemma A.3), the graph convoluted label (A.4), the graph convoluted noise vector (Lemma A.7 and Lemma A.5) are no longer guaranteed. This lack of concentration affects the behavior of coefficients during gradient descent training, leading to deviations from our current main results.

Lemma A.3 (Degree concentration)**.** *Let $p, s = \Omega\left(\sqrt{\frac{\log(n/\delta)}{n}}\right)$ and $\delta > 0$, then with probability at least $1 - \delta$, we have*

$$n(p+s)/4 \leq D_i \leq 3n(p+s)/4.$$

Proof. It is known that the degrees are sums of Bernoulli random variables.

$$D_i = 1 + \sum_{j \neq i}^{n} a_{ij},$$

where $a_{ij} = [\mathbf{A}]_{ij}$. Hence, by the Hoeffding's inequality, with probability at least $1 - \delta/n$

$$|D_i - \mathbb{E}[D_i]| < \sqrt{\log(n/\delta)(n-1)}.$$

Note that $a_{ii} = 1$ is a fixed value, which means that it is not a random variable, thus the denominator in the exponential part is $n - 1$ instead of n. Now we calculate the expectation of degree:

$$\mathbb{E}[D_{ii}] = 1 + \frac{n}{2}s + (\frac{n}{2} - 1)p = n(p+s)/2 + 1 - p,$$

then we have

$$|D_i - n(p+s)/2 + 1 - p| \leq \sqrt{n\log(n/\delta)}.$$

Because that $p, s = \Omega\left(\sqrt{\frac{\log(n/\delta)}{n}}\right)$, we further have,

$$n(p+s)/4 \leq D_i \leq 3n(p+s)/4.$$

Applying a union bound over $i \in [n]$ conclude the proof. □

Lemma A.4. *Suppose that $\delta > 0$ and $n \geq 8\frac{p+s}{(p-s)^2}\log(4/\delta)$. Then with probability at least $1-\delta$,*

$$\frac{1}{2}\frac{p-s}{p+s}|y_i| \leq |\tilde{y}_i| \leq \frac{3}{2}\frac{p-s}{p+s}|y_i|.$$

Proof of Lemma A.4. By Hoeffding's inequality, with probability at least $1-\delta/2$, we have

$$\left|\frac{1}{D_i}\sum_{k\in\mathcal{N}(i)} y_k - \frac{p-s}{p+s}y_i\right| \leq \sqrt{\frac{\log(4/\delta)}{2n(p+s)}}.$$

Therefore, as long as $n \geq 8\frac{p+s}{(p-s)^2}\log(4/\delta)$, we have:

$$\frac{1}{2}\frac{p-s}{p+s}|y_i| \leq |\tilde{y}_i| \leq \frac{3}{2}\frac{p-s}{p+s}|y_i|.$$

This proves the result for the stability of sign of graph convoluted label. □

Lemma A.5. *Suppose that $\delta > 0$ and $d = \Omega(n^2(p+s)^2\log(4n^2/\delta))$. Then with probability at least $1-\delta$,*

$$\sigma_p^2 d/(4n(p+s)) \leq \|\tilde{\boldsymbol{\xi}}_i\|_2^2 \leq 3\sigma_p^2 d/(4n(p+s)),$$

for all $i \in [n]$.

Proof of Lemma A.5. It is known that:

$$\|\tilde{\boldsymbol{\xi}}_i\|_2^2 = \frac{1}{D_i^2}\sum_{j=1}^{d}\left(\sum_{k=1}^{D_i}\xi_{jk}\right)^2 = \frac{1}{D_i^2}\sum_{j=1}^{d}\sum_{k=1}^{D_i}\xi_{jk}^2 + \frac{1}{D_i^2}\sum_{j=1}^{d}\sum_{k\neq k'}^{D_i}\xi_{jk'}\xi_{jk}.$$

By Bernstein's inequality, with probability at least $1-\delta/(2n)$ we have

$$\left|\sum_{j=1}^{d}\sum_{k=1}^{D_i}\xi_{jk}^2 - \sigma_p^2 dD_i\right| = O(\sigma_p^2 \cdot \sqrt{dD_i\log(4n/\delta)}).$$

Therefore, as long as $d = \Omega(\log(4n/\delta)/(n(p+s)))$, we have

$$3\sigma_p^2 dD_i/4 \leq \sum_{j=1}^{d}\sum_{k=1}^{D_i}\xi_{jk}^2 \leq 5\sigma_p^2 dD_i/4.$$

By Lemma A.3, we have,

$$2\sigma_p^2 d/(4n(p+s)) \leq \frac{1}{D_i^2}\sum_{j=1}^{d}\sum_{k=1}^{D_i}\xi_{jk}^2 \leq 6\sigma_p^2 d/(4n(p+s)).$$

Moreover, clearly $\langle \boldsymbol{\xi}_k, \boldsymbol{\xi}_{k'} \rangle$ has mean zero. For any k, k' with $k \neq k'$, by Bernstein's inequality, with probability at least $1 - \delta/(2n^2)$ we have

$$|\langle \boldsymbol{\xi}_k, \boldsymbol{\xi}_{k'} \rangle| \leq 2\sigma_p^2 \cdot \sqrt{d \log(4n^2/\delta)}.$$

Applying a union bound we have that with probability at least $1 - \delta$,

$$|\langle \boldsymbol{\xi}_k, \boldsymbol{\xi}_{k'} \rangle| \leq 2\sigma_p^2 \cdot \sqrt{d \log(4n^2/\delta)}.$$

Therefore, as long as $d = \Omega(n^2(p+s)^2 \log(4n^2/\delta))$, we have

$$\sigma_p^2 d/(4n(p+s)) \leq \|\tilde{\boldsymbol{\xi}}_i\|_2^2 \leq 3\sigma_p^2 d/(4n(p+s)).$$

Remark A.6. We compare the noise vector both before and after applying graph convolution. By examining Lemma A.1 and Lemma A.5, we discover that the expectation of the ℓ_2 norm of noise vector is reduced by a factor of $\sqrt{n(p+s)/2}$. This factor represents the square root of the expected degree of the graph, indicating a significant change in the noise characteristics as a result of the graph convolution process. We provide a demonstrative visualization in Figure 7.

□

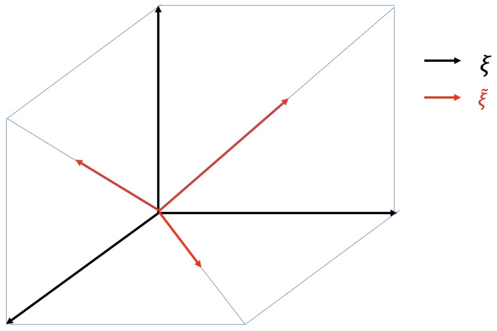


Figure 7: An illustrative example of noise vector before and after graph aggregation. In this example, we consider $d = 3$ and all degree are 1. The black vectors stand for noise vectors $\boldsymbol{\xi}$ before graph convolution. Each of them are orthogonal to each other. The red vectors represent noise vectors after graph convolution $\tilde{\boldsymbol{\xi}}$. They are graph convoluted noise vectors of two original noise vectors. Note that the ℓ_2 norm between two kinds of vector follows $\|\tilde{\boldsymbol{\xi}}\|_2 = \frac{\sqrt{2}}{2}\|\boldsymbol{\xi}\|_2$. This plot demonstrates how graph convolution shrinks the ℓ_2 norm of noise vectors.

Lemma A.7. *Suppose that $d = \Omega(n(p+s)\log(nm/\delta))$, $m = \Omega(\log(1/\delta))$. Then with probability at least $1 - \delta$,*

$$|\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i \rangle| \leq 4\sqrt{\log(8mn/\delta)} \cdot \sigma_0 \sigma_p \sqrt{d/(n(p+s))},$$
$$\sigma_0 \sigma_p \sqrt{d/(n(p+s))}/4 \leq \max_{r \in [m]} j \cdot \langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i \rangle \leq 2\sqrt{\log(8mn/\delta)} \cdot \sigma_0 \sigma_p \sqrt{d/(n(p+s))},$$

for all $j \in \{\pm 1\}$ and $i \in [n]$.

Proof of Lemma A.7. According to the fact that the weight $\mathbf{w}_{j,r}(0)$ and noise vector $\boldsymbol{\xi}$ are sampled from Gaussian distribution, we know that $\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i \rangle$ is also Gaussian. By Lemma A.5, with probability at least $1 - \delta/4$, we have that

$$\sigma_p \sqrt{d/(n(p+s))}/\sqrt{2} \leq \|\tilde{\boldsymbol{\xi}}_i\|_2 \leq \sqrt{3/2} \cdot \sigma_p \sqrt{d/(n(p+s))}$$

holds for all $i \in [n]$. Therefore, applying the concentration bound for Gaussian variable, we obtain that

$$|\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i \rangle| \leq 4\sqrt{\log(8mn/\delta)} \cdot \sigma_0 \sigma_p \sqrt{d/(n(p+s))}.$$

Next we finish the argument for the lower bound of maximum through the follow expression:

$$
\begin{aligned}
P(\max\langle \mathbf{w}^{(0)}_{j,r}, \tilde{\boldsymbol{\xi}}_i\rangle \geq \sigma_0\sigma_p\sqrt{d/(n(p+s))}/4) &= 1 - P(\max\langle \mathbf{w}^{(0)}_{j,r}, \tilde{\boldsymbol{\xi}}_i\rangle < \sigma_0\sigma_p\sqrt{d/(n(p+s))}/4) \\
&= 1 - P(\max\langle \mathbf{w}^{(0)}_{j,r}, \tilde{\boldsymbol{\xi}}_i\rangle < \sigma_0\sigma_p\sqrt{d/(n(p+s))}/4)^{2m} \\
&\geq 1 - \delta/4.
\end{aligned}
$$

Together with Lemma A.5, we finally obtain that

$$
\sigma_0\sigma_p\sqrt{d/(n(p+s))}/4 \leq \max_{r\in[m]} j \cdot \langle \mathbf{w}^{(0)}_{j,r}, \tilde{\boldsymbol{\xi}}_i\rangle \leq 2\sqrt{\log(8mn/\delta)} \cdot \sigma_0\sigma_p\sqrt{d/(n(p+s))}.
$$

□

B General Lemmas for Iterative Coefficient Analysis

In this section, we deliver lemmas that delineate the iterative behavior of coefficients under gradient descent. We commence with proving the coefficient update rules as stated in Lemma 5.1 in Section B.1. Subsequently, we establish the scale of training dynamics in Section B.2.

B.1 Coefficient update rule

Lemma B.1 (Restatement of Lemma 5.1). *The coefficients $\gamma^{(t)}_{j,r}, \zeta^{(t)}_{j,r,i}, \omega^{(t)}_{j,r,i}$ defined in Eq. (6) satisfy the following iterative equations:*

$$
\begin{aligned}
&\gamma^{(0)}_{j,r}, \overline{\rho}^{(0)}_{j,r,i}, \underline{\rho}^{(0)}_{j,r,i} = 0, \\
&\gamma^{(t+1)}_{j,r} = \gamma^{(t)}_{j,r} - \frac{\eta}{nm} \cdot \sum_{i=1}^{n} \ell'^{(t)}_i \sigma'(\langle \mathbf{w}^{(t)}_{j,r}, \tilde{y}_i \boldsymbol{\mu}\rangle) y_i \tilde{y}_i \|\boldsymbol{\mu}\|_2^2, \\
&\overline{\rho}^{(t+1)}_{j,r,i} = \overline{\rho}^{(t)}_{j,r,i} - \frac{\eta}{nm} \cdot \sum_{k\in\mathcal{N}(i)} D_k^{-1} \cdot \ell'^{(t)}_k \cdot \sigma'(\langle \mathbf{w}^{(t)}_{j,r}, \tilde{\boldsymbol{\xi}}_k\rangle) \cdot \|\boldsymbol{\xi}_i\|_2^2 \cdot \mathbb{1}(y_k = j), \\
&\underline{\rho}^{(t+1)}_{j,r,i} = \underline{\rho}^{(t)}_{j,r,i} - \frac{\eta}{nm} \cdot \sum_{k\in\mathcal{N}(i)} D_k^{-1} \cdot \ell'^{(t)}_k \cdot \sigma'(\langle \mathbf{w}^{(t)}_{j,r}, \tilde{\boldsymbol{\xi}}_k\rangle) \cdot \|\boldsymbol{\xi}_i\|_2^2 \cdot \mathbb{1}(y_k = -j),
\end{aligned}
$$

for all $r \in [m]$, $j \in \{\pm 1\}$ and $i \in [n]$.

Remark B.2. This lemma serves as a foundational element in our analysis of dynamics. Initially, the study of neural network dynamics under gradient descent required us to monitor the fluctuations in weights. However, this Lemma enables us to observe these dynamics through a new lens, focusing on two distinct aspects: signal learning and noise memorization. These are represented by the variables $\gamma^{(t)}_{j,r}$ and $\rho^{(t)}_{j,r,i}$, respectively. Furthermore, the selection of our data model was a conscious decision, designed to clearly separate the signal learning from the noise memorization aspects of learning. By maintaining a clear distinction between signal and noise, we can conduct a precise analysis of how each model learns the signal and memorizes the noise. This approach not only simplifies our understanding but also enhances our ability to dissect the underlying mechanisms of learning.

Proof of Lemma B.1. Basically, the iteration of coefficients is derived based on gradient descent rule (4) and weight decomposition (6). We first consider $\hat{\gamma}^{(0)}_{j,r}, \hat{\rho}^{(0)}_{j,r,i} = 0$ and

$$
\begin{aligned}
\hat{\gamma}^{(t+1)}_{j,r} &= \hat{\gamma}^{(t)}_{j,r} - \frac{\eta}{nm} \cdot \sum_{i=1}^{n} \ell'^{(t)}_i \sigma'(\langle \mathbf{w}^{(t)}_{j,r}, \tilde{y}_i \boldsymbol{\mu}_i\rangle) y_i \tilde{y}_i \|\boldsymbol{\mu}\|_2^2, \\
\hat{\rho}^{(t+1)}_{j,r,i} &= \hat{\rho}^{(t)}_{j,r,i} - \frac{\eta}{nm} \cdot \sum_{k\in\mathcal{N}(i)} D_k^{-1} \cdot \ell'^{(t)}_k \cdot \sigma'(\langle \mathbf{w}^{(t)}_{j,r}, \tilde{\boldsymbol{\xi}}_k\rangle) \cdot \|\boldsymbol{\xi}_i\|_2^2 \cdot y_k,
\end{aligned}
$$

Taking above equations into Equation (4), we can obtain that

$$
\mathbf{w}^{(t)}_{j,r} = \mathbf{w}^{(0)}_{j,r} + j \cdot \hat{\gamma}^{(t)}_{j,r} \cdot \|\boldsymbol{\mu}\|_2^{-2} \cdot \boldsymbol{\mu} + \sum_{i=1}^{n} \hat{\rho}^{(t)}_{j,r,i} \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i.
$$

This result verifies that the iterative update of the coefficients is directly driven by the gradient descent update process. Furthermore, the uniqueness of the decomposition leads us to the precise relationships $\gamma_{j,r}^{(t)} = \hat{\gamma}_{j,r}^{(t)}$ and $\rho_{j,r,i}^{(t)} = \hat{\rho}_{j,r,i}^{(t)}$. Next, we examine the stability of the sign associated with noise memorization by employing the following telescopic analysis. This method allows us to investigate the continuity and consistency of the noise memorization process, providing insights into how the system behaves over successive iterations.

$$\rho_{j,r,i}^{(t)} = -\sum_{s=0}^{t-1} \sum_{k \in \mathcal{N}(i)} D_k^{-1} \frac{\eta}{nm} \cdot \ell_k'^{(s)} \cdot \sigma'(\langle \mathbf{w}_{j,r}^{(s)}, \tilde{\boldsymbol{\xi}}_k \rangle) \cdot \|\boldsymbol{\xi}_i\|_2^2 \cdot j y_k.$$

Recall the sign of loss derivative is given by the definition of the cross-entropy loss, namely, $\ell_i'^{(t)} < 0$. Therefore,

$$\overline{\rho}_{j,r,i}^{(t)} = -\sum_{s=0}^{t-1} \frac{\eta}{nm} \cdot \sum_{k \in \mathcal{N}(i)} D_k^{-1} \cdot \ell_k'^{(t)} \cdot \sigma'(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_k \rangle) \cdot \|\boldsymbol{\xi}_i\|_2^2 \cdot \mathbb{1}(y_k = j), \tag{10}$$

$$\underline{\rho}_{j,r,i}^{(t)} = -\sum_{s=0}^{t-1} \frac{\eta}{nm} \cdot \sum_{k \in \mathcal{N}(i)} D_k^{-1} \cdot \ell_k'^{(t)} \cdot \sigma'(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_k \rangle) \cdot \|\boldsymbol{\xi}_i\|_2^2 \cdot \mathbb{1}(y_k = -j). \tag{11}$$

Writing out the iterative versions of (10) and (11) completes the proof. □

Remark B.3. The proof strategy follows the study of feature learning in CNN as described in [8]. However, compared to CNNs, the decomposition of weights in GNN is notably more intricate. This complexity is particularly evident in the dynamics of noise memorization, as represented by Equations 10) and 11). The reason for this increased complexity lies in the additional graph convolution operations within GNNs. These operations introduce new interaction and dependencies, making the analysis of weight dynamics more challenging and nuanced.

B.2 Scale of training dynamics

Our proof hinges on a meticulous evaluation of the coefficient values $\gamma_{j,r}^{(t)}, \overline{\rho}_{j,r,i}^{(t)}, \underline{\rho}_{j,r,i}^{(t)}$ throughout the entire training process. In order to facilitate a more thorough analysis, we first establish the following bounds for these coefficients, which are maintained consistently throughout the training period.

Consider training the Graph Neural Network (GNN) for an extended period up to T^*. We aim to investigate the scale of noise memorization in relation to signal learning.

Let $T^* = \eta^{-1}\mathrm{poly}(\epsilon^{-1}, \|\boldsymbol{\mu}\|_2^{-1}, d^{-1}\sigma_p^{-2}, \sigma_0^{-1}, n, m, d)$ be the maximum admissible iterations. Denote $\alpha = 4\log(T^*)$. In preparation for an in-depth analysis, we enumerate the necessary conditions that must be satisfied. These conditions, which are essential for the subsequent examination, are also detailed in Condition 4.1:

$$\eta = O\Big(\min\{nm/(q\sigma_p^2 d), nm/(q2^{q+2}\alpha^{q-2}\sigma_p^2 d), nm/(q2^{q+2}\alpha^{q-2}\|\boldsymbol{\mu}\|_2^2)\}\Big), \tag{12}$$

$$\sigma_0 \le [16\sqrt{\log(8mn/\delta)}]^{-1} \min\left\{\Xi^{-1}\|\boldsymbol{\mu}\|_2^{-1}, (\sigma_p\sqrt{d/(n(p+s))})^{-1}\right\}, \tag{13}$$

$$d \ge 1024\log(4n^2/\delta)\alpha^2 n^2. \tag{14}$$

Denote $\beta = 2\max_{i,j,r}\{|\langle \mathbf{w}_{j,r}^{(0)}, \tilde{y}_i \cdot \boldsymbol{\mu}\rangle|, |\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i\rangle|\}$, it is straightforward to show the following inequality:

$$4\max\left\{\beta, 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha\right\} \le 1. \tag{15}$$

First, by Lemma A.4 with probability at least $1-\delta$, we can upper bound β by $4\sqrt{\log(8mn/\delta)} \cdot \sigma_0 \cdot \max\{\Xi\|\boldsymbol{\mu}\|_2, \sigma_p\sqrt{d/(n(p+s))}\}$. Combined with the condition (13), we can bound β by 1. Second, it is easy to check that $8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha \le 1$ by inequality (14).

Having established the values of α and β at hand, we are now in a position to assert that the following proposition holds for the entire duration of the training process, specifically for $0 \le t \le T^*$.

Proposition B.4. *Under Condition 4.1, for $0 \le t \le T^*$, where $T^* = \eta^{-1}\mathrm{poly}(\epsilon^{-1}, \|\boldsymbol{\mu}\|_2^{-1}, d^{-1}\sigma_p^{-2}, \sigma_0^{-1}, n, m, d)$, we have that*

$$0 \le \gamma_{j,r}^{(t)}, \overline{\rho}_{j,r,i}^{(t)} \le \alpha, \tag{16}$$

$$0 \ge \underline{\rho}_{j,r,i}^{(t)} \ge -\alpha, \tag{17}$$

for all $r \in [m]$, $j \in \{\pm 1\}$ and $i \in [n]$, where $\alpha = 4\log(T^*)$.

To establish Proposition B.4, we will employ an inductive approach. Before proceeding with the proof, we need to introduce several technical lemmas that are fundamental to our argument.

We note that although the setting is slightly different from the case in [8]. With the same analysis, we can obtain the following result.

Lemma B.5 ([8])**.** *For any $t \ge 0$, it holds that $\langle \mathbf{w}_{j,r}^{(t)} - \mathbf{w}_{j,r}^{(0)}, \boldsymbol{\mu}\rangle = j \cdot \gamma_{j,r}^{(t)}$ for all $r \in [m]$, $j \in \{\pm 1\}$.*

In the subsequent three lemmas, our proof strategy is guided by the approach found in [8]. However, we extend this methodology by providing a fine-grained analysis that takes into account the additional complexity introduced by the graph convolution operation.

Lemma B.6. *Under Condition 4.1, suppose* (16) *and* (17) *hold at iteration t. Then*

$$\hat{\rho}_{j,r,i}^{(t)} - 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha \le \langle \mathbf{w}_{j,r}^{(t)} - \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i\rangle \le \hat{\rho}_{j,r,i}^{(t)} + 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha,$$

where $\hat{\rho}_{j,r,i} \triangleq \sum_{k\in\mathcal{N}(i)} D_i^{-1}\sum_{i'\neq k}\rho_{j,r,i'}^{(t)}$, for all $r \in [m]$, $j \in \{\pm 1\}$ and $i \in [n]$.

Remark B.7. Lemma B.6 asserts that the inner product between the updated weight and the graph convolution operation closely approximates the graph-convoluted noise memorization.

Proof of Lemma B.6. It is known that,

$$\begin{aligned}
\langle \mathbf{w}_{j,r}^{(t)} - \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i\rangle &= \sum_{i'=1}^{n} \zeta_{j,r,i'}^{(t)} \|\boldsymbol{\xi}_{i'}\|_2^{-2} \cdot \langle \boldsymbol{\xi}_{i'}, \tilde{\boldsymbol{\xi}}_i\rangle + \sum_{i'=1}^{n} \omega_{j,r,i'}^{(t)} \|\boldsymbol{\xi}_{i'}\|_2^{-2} \cdot \langle \boldsymbol{\xi}_{i'}, \tilde{\boldsymbol{\xi}}_i\rangle \\
&= \sum_{i'=1}^{n}\sum_{k\in\mathcal{N}(i)} D_i^{-1}\zeta_{j,r,i'}^{(t)} \|\boldsymbol{\xi}_{i'}\|_2^{-2} \cdot \langle \boldsymbol{\xi}_{i'}, \boldsymbol{\xi}_k\rangle + \sum_{i'=1}^{n}\sum_{k\in\mathcal{N}(i)} D_i^{-1}\omega_{j,r,i'}^{(t)} \|\boldsymbol{\xi}_{i'}\|_2^{-2} \cdot \langle \boldsymbol{\xi}_{i'}, \boldsymbol{\xi}_k\rangle \\
&\le 4\sqrt{\frac{\log(4n^2/\delta)}{d}} \sum_{k\in\mathcal{N}(i)} D_i^{-1}\sum_{i'\neq k}|\zeta_{j,r,i'}^{(t)}| + 4\sqrt{\frac{\log(4n^2/\delta)}{d}} \sum_{k\in\mathcal{N}(i)} D_i^{-1}\sum_{i'\neq k}|\omega_{j,r,i'}^{(t)}| \\
&\quad + \sum_{k\in\mathcal{N}(i)} D_i^{-1}\sum_{i'\neq k}\zeta_{j,r,i'}^{(t)} + \sum_{k\in\mathcal{N}(i)} D_i^{-1}\sum_{i'\neq k}\omega_{j,r,i'}^{(t)} \\
&\le \hat{\rho}_{j,r,i}^{(t)} + 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha,
\end{aligned}$$

where we define $\hat{\rho}_{j,r,i} \triangleq \sum_{k\in\mathcal{N}(i)} D_i^{-1}\sum_{i'\neq k}\rho_{j,r,i'}^{(t)}$ the second inequality is by Lemma A.1 and the last inequality is by $|\zeta_{j,r,i'}^{(t)}|, |\omega_{j,r,i'}^{(t)}| \le \alpha$ in (16).

Similarly, we can show that:

$$\begin{aligned}
\langle \mathbf{w}_{j,r}^{(t)} - \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i \rangle &= \sum_{i'=1}^{n} \zeta_{j,r,i'}^{(t)} \|\boldsymbol{\xi}_{i'}\|_2^{-2} \cdot \langle \boldsymbol{\xi}_{i'}, \tilde{\boldsymbol{\xi}}_i \rangle + \sum_{i'=1}^{n} \omega_{j,r,i'}^{(t)} \|\boldsymbol{\xi}_{i'}\|_2^{-2} \cdot \langle \boldsymbol{\xi}_{i'}, \tilde{\boldsymbol{\xi}}_i \rangle \\
&= \sum_{i'=1}^{n} \sum_{k \in \mathcal{N}(i)} D_i^{-1} \zeta_{j,r,i'}^{(t)} \|\boldsymbol{\xi}_{i'}\|_2^{-2} \cdot \langle \boldsymbol{\xi}_{i'}, \boldsymbol{\xi}_k \rangle + \sum_{i'=1}^{n} \sum_{k \in \mathcal{N}(i)} D_i^{-1} \omega_{j,r,i'}^{(t)} \|\boldsymbol{\xi}_{i'}\|_2^{-2} \cdot \langle \boldsymbol{\xi}_{i'}, \boldsymbol{\xi}_k \rangle \\
&\geq -4\sqrt{\frac{\log(4n^2/\delta)}{d}} \sum_{k \in \mathcal{N}(i)} D_i^{-1} \sum_{i' \neq k} |\zeta_{j,r,i'}^{(t)}| - 4\sqrt{\frac{\log(4n^2/\delta)}{d}} \sum_{k \in \mathcal{N}(i)} D_i^{-1} \sum_{i' \neq k} |\omega_{j,r,i'}^{(t)}| \\
&\quad + \sum_{k \in \mathcal{N}(i)} D_i^{-1} \sum_{i' \neq k} \zeta_{j,r,i'}^{(t)} + \sum_{k \in \mathcal{N}(i)} D_i^{-1} \sum_{i' \neq k} \omega_{j,r,i'}^{(t)} \\
&\geq \hat{\rho}_{j,r,i}^{(t)} - 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha,
\end{aligned}$$

where the first inequality is by Lemma A.1 and the second inequality is by $|\zeta_{j,r,i'}^{(t)}|, |\omega_{j,r,i'}^{(t)}| \leq \alpha$ in (16), which completes the proof. □

Lemma B.8. *Under Condition 4.1, suppose* (16) *and* (17) *hold at iteration* t. *Then*

$$\begin{aligned}
\langle \mathbf{w}_{j,r}^{(t)}, \tilde{y}_i \boldsymbol{\mu} \rangle &\leq \langle \mathbf{w}_{j,r}^{(0)}, \tilde{y}_i \boldsymbol{\mu} \rangle, \\
\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_i \rangle &\leq \langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i \rangle + 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha,
\end{aligned}$$

for all $r \in [m]$ *and* $j \neq y_i$. *If* $\max\{\gamma_{j,r}^{(t)}, \rho_{j,r,i}^{(t)}\} = O(1)$, *we further have that* $F_j(\mathbf{W}_j^{(t)}, \tilde{\mathbf{x}}_i) = O(1)$.

Remark B.9. Lemma B.8 further establishes that the update in the direction of $\tilde{\boldsymbol{\xi}}$ can be constrained within specific bounds when $j \neq y_i$. As a result, the output function remains controlled and does not exceed a constant order.

Proof of Lemma B.8. For $j \neq y_i$, we have that

$$\langle \mathbf{w}_{j,r}^{(t)}, \tilde{y}_i \boldsymbol{\mu} \rangle = \langle \mathbf{w}_{j,r}^{(0)}, \tilde{y}_i \boldsymbol{\mu} \rangle + \tilde{y}_i \cdot j \cdot \gamma_{j,r}^{(t)} \leq \langle \mathbf{w}_{j,r}^{(0)}, \tilde{y}_i \boldsymbol{\mu} \rangle, \tag{18}$$

where the inequality is by $\gamma_{j,r}^{(t)} \geq 0$ and Lemma A.4 stating that $\mathrm{sign}(y_i) = \mathrm{sign}(\tilde{y}_i)$ with a high probability. In addition, we have

$$\begin{aligned}
\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_i \rangle &= \langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i \rangle + \sum_{k \in \mathcal{N}(i)} D_i^{-1} \sum_{i'=1}^{n} \rho_{j,r,i'} \langle \boldsymbol{\xi}_k, \boldsymbol{\xi}_{i'} \rangle \|\boldsymbol{\xi}_{i'}\|_2^{-2} \\
&\leq \langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i \rangle + D_i^{-1} \left(\sum_{y_k \neq j} \omega_{j,r,i}^{(t)} + \sum_{y_k = j} \zeta_{j,r,i}^{(t)} \right) + 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha \\
&\leq \langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i \rangle + 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha, \qquad (19)
\end{aligned}$$

where the first inequality is by Lemma B.6 and the second inequality is due to $\hat{\rho}_{j,r,i}^{(t)} \leq 0$ based on Lemma A.4.

Then we can get that

$$\begin{aligned}
F_j(\mathbf{W}_j^{(t)}, \tilde{\mathbf{x}}_i) &= \frac{1}{m}\sum_{r=1}^{m}[\sigma(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{y}_i \cdot \boldsymbol{\mu}\rangle) + \sigma(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_i\rangle)] \\
&= \frac{1}{m}\sum_{r=1}^{m}[\sigma(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{y}_i \cdot \boldsymbol{\mu}\rangle) + \sigma(\langle \mathbf{w}_{j,r}^{(t)}, D_i^{-1}\sum_{k\in\mathcal{N}(i)} \boldsymbol{\xi}_k\rangle)] \\
&= \frac{1}{m}\sum_{r=1}^{m}[\sigma(\langle \mathbf{w}_{j,r}^{(0)}, \tilde{y}_i \cdot \boldsymbol{\mu}\rangle) + \sigma(\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i\rangle + \langle \mathbf{w}_{j,r}^{(t)} - \mathbf{w}_{j,r}^{(0)}, D_i^{-1}\sum_{k\in\mathcal{N}(i)} \boldsymbol{\xi}_k\rangle)] \\
&\le \frac{1}{m}\sum_{r=1}^{m}[\sigma(\langle \mathbf{w}_{j,r}^{(0)}, \tilde{y}_i \cdot \boldsymbol{\mu}\rangle) + \sigma(\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i\rangle + 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha + \hat{\rho}_{j,r,i}^{(t)})] \\
&\le 2^{q+1}\max_{j,r,i}\left\{|\langle \mathbf{w}_{j,r}^{(0)}, \tilde{y}_i \cdot \boldsymbol{\mu}\rangle|, |\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i\rangle|, 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha\right\}^q \\
&\le 1,
\end{aligned}$$

where the first inequality is by (18), (19) and the second inequality is by (15) and $\max\{\gamma_{j,r}^{(t)}, \rho_{j,r,i}^{(t)}\} = O(1)$. □

Lemma B.10. *Under Condition 4.1, suppose* (16) *and* (17) *hold at iteration* t. *Then*

$$\begin{aligned}
\langle \mathbf{w}_{j,r}^{(t)}, \tilde{y}_i\boldsymbol{\mu}\rangle &= \langle \mathbf{w}_{j,r}^{(0)}, \tilde{y}_i\boldsymbol{\mu}\rangle + \gamma_{j,r}^{(t)}, \\
\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_i\rangle &\le \langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i\rangle + \hat{\rho}_{j,r,i}^{(t)} + 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha
\end{aligned}$$

for all $r \in [m]$, $j = y_i$ *and* $i \in [n]$. *If* $\max\{\gamma_{j,r}^{(t)}, \rho_{j,r,i}^{(t)}\} = O(1)$, *we further have that* $F_j(\mathbf{W}_j^{(t)}, \tilde{\mathbf{x}}_i) = O(1)$.

Remark B.11. Lemma B.10 further establishes that the update in the direction of $\boldsymbol{\mu}$ and $\tilde{\boldsymbol{\xi}}$ can be constrained within specific bounds when $j = y_i$. As a result, the output function remains controlled and does not exceed a constant order with an additional condition.

Proof of Lemma B.10. For $j = y_i$, we have that

$$\langle \mathbf{w}_{j,r}^{(t)}, \tilde{y}_i\boldsymbol{\mu}\rangle = \langle \mathbf{w}_{j,r}^{(0)}, \tilde{y}_i\boldsymbol{\mu}\rangle + \gamma_{j,r}^{(t)}, \tag{20}$$

where the equation is by Lemma B.5. We also have that

$$\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_i\rangle \le \langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i\rangle + \hat{\rho}_{j,r,i}^{(t)} + 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha, \tag{21}$$

where the inequality is by Lemma B.6. If $\max\{\gamma_{j,r}^{(t)}, \rho_{j,r,i}^{(t)}\} = O(1)$, we have following bound

$$\begin{aligned}
F_j(\mathbf{W}_j^{(t)}, \tilde{\mathbf{x}}_i) &= \frac{1}{m}\sum_{r=1}^{m}[\sigma(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{y}_i \cdot \boldsymbol{\mu}) + \sigma(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_i\rangle)] \\
&\le 2 \cdot 3^q \max_{j,r,i}\left\{\gamma_{j,r}^{(t)}, |\hat{\rho}_{j,r,i}^{(t)}|, |\langle \mathbf{w}_{j,r}^{(0)}, \tilde{y}_i \cdot \boldsymbol{\mu})\rangle|, |\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i\rangle|, 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha\right\}^q \\
&= O(1),
\end{aligned}$$

where $\hat{\rho}_{j,r,i}^{(t)} = \frac{1}{D_i}\sum_{k\in\mathcal{N}(i)} \overline{\rho}_{j,r,k}^{(t)}\mathbb{1}(y_k = j) + \overline{\rho}_{j,r,k}^{(t)}\mathbb{1}(y_k \neq j)$, the first inequality is by (20), (21). Then the second inequality is by (15) where $\beta = 2\max_{i,j,r}\{|\langle \mathbf{w}_{j,r}^{(0)}, \tilde{y}_i \cdot \boldsymbol{\mu}\rangle|, |\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i\rangle|\} \le 1$ and condition that $\max\{\gamma_{j,r}^{(t)}, \rho_{j,r,i}^{(t)}\} = O(1)$. □

Equipped with Lemmas B.5 - B.10, we are now prepared to prove Proposition B.4. These lemmas provide the foundational building blocks and insights necessary for our proof, setting the stage for a rigorous and comprehensive demonstration of the proposition

Proof of Proposition B.4. Following a similar approach to the proof found in [8], we employ an induction method. This technique allows us to build our argument step by step, drawing on established principles and extending them to our specific context, thereby providing a robust and systematic demonstration.

At the initial time step $t = 0$, the outcome is clear since all coefficients are set to zero.

Next, we hypothesize that there exists a time $\tilde{T}$ less that T^* during which Proposition B.4 holds true for every moment within the range $0 \le t \le \tilde{T} - 1$. Our objective is to show that this proposition remains valid at $t = \tilde{T}$.

We aim to validate that equation (17) is applicable at $t = \tilde{T}$, meaning that,

$$\omega_{j,r,i}^{(t)} \ge -\beta - 16n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha,$$

for the given parameters. It's important to note that $\omega_{j,r,i}^{(t)} = 0$ when $j = y_i$. So we only need to consider instances where $j \ne y_i$.

1) Under condition

$$\omega_{j,r,i}^{(\tilde{T}-1)} \le -0.5\beta - 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha,$$

Lemma B.6 leads us to the following relationships:

$$\langle \mathbf{w}_{j,r}^{(\tilde{T}-1)}, \tilde{y}_i\boldsymbol{\mu}\rangle \le \hat{\rho}_{j,r,i}^{(\tilde{T}-1)} + \langle \mathbf{w}_{j,r}^{(0)}, \tilde{y}_i\boldsymbol{\mu}\rangle + 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha \le 0,$$

and thus

$$\begin{aligned}\omega_{j,r,i}^{(\tilde{T})} &= \omega_{j,r,i}^{(\tilde{T}-1)} + \frac{\eta}{nm}\sum_{k} D_k^{-1} \cdot \ell_k'^{(\tilde{T}-1)} \cdot \sigma'(\langle \mathbf{w}_{j,r}^{(\tilde{T}-1)}, \tilde{\boldsymbol{\xi}}_k\rangle) \cdot \mathbb{1}(y_k = -j)\|\boldsymbol{\xi}_i\|_2^2 \\ &= \omega_{j,r,i}^{(\tilde{T}-1)} \ge -\beta - 16n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha,\end{aligned}$$

with the final inequality being supported by the induction hypothesis.

2) Given the condition $\omega_{j,r,i}^{(\tilde{T}-1)} \ge -0.5\beta - 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha$, we can derive the following:

$$\begin{aligned}\omega_{j,r,i}^{(\tilde{T})} &= \omega_{j,r,i}^{(\tilde{T}-1)} + \frac{\eta}{nm} \cdot \sum_{k\in\mathcal{N}(i)} D_k^{-1}\ell_k'^{(\tilde{T}-1)} \cdot \sigma'(\langle \mathbf{w}_{j,r}^{(T-1)}, \tilde{\boldsymbol{\xi}}_k\rangle) \cdot \mathbb{1}(y_k = -j)\|\boldsymbol{\xi}_i\|_2^2 \\ &\ge -0.5\beta - 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha - O\Big(\frac{\eta\sigma_p^2 d}{nm}\Big)\sigma'\Big(0.5\beta + 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha\Big) \\ &\ge -0.5\beta - 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha - O\Big(\frac{\eta q\sigma_p^2 d}{nm}\Big)\Big(0.5\beta + 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha\Big) \\ &\ge -\beta - 16n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha,\end{aligned}$$

where we apply the inequalities $\ell_i'^{(\tilde{T}-1)} \le 1$ and $\|\boldsymbol{\xi}_i\|_2 = O(\sigma_p^2 d)$, and use the conditions $\eta = O\big(nm/(q\sigma_p^2 d)\big)$ and $0.5\beta + 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha \le 1$, as specified in (12).

Next, we aim to show that (16) is valid for $t = \tilde{T}$. We can express:

$$\begin{aligned}|\ell_i'^{(t)}| &= \frac{1}{1 + \exp\{y_i \cdot [F_{+1}(\mathbf{W}_{+1}^{(t)}, \tilde{\mathbf{x}}_i) - F_{-1}(\mathbf{W}_{-1}^{(t)}, \tilde{\mathbf{x}}_i)]\}} \\ &\le \exp\{-y_i \cdot [F_{+1}(\mathbf{W}_{+1}^{(t)}, \tilde{\mathbf{x}}_i) - F_{-1}(\mathbf{W}_{-1}^{(t)}, \tilde{\mathbf{x}}_i)]\} \\ &\le \exp\{-F_{y_i}(\mathbf{W}_{y_i}^{(t)}, \tilde{\mathbf{x}}_i) + 1\}. \qquad (22)\end{aligned}$$

with the last inequality being a result of Lemma B.8. Additionally, we recall the update rules for $\gamma_{j,r}^{(t+1)}$ and $\zeta_{j,r,i}^{(t+1)}$:

$$\gamma_{j,r}^{(t+1)} = \gamma_{j,r}^{(t)} - \frac{\eta}{nm} \cdot \sum_{i=1}^{n} \ell_i'^{(t)} \cdot \sigma'(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{y}_i \cdot \boldsymbol{\mu} \rangle) y_i \tilde{y}_i \|\boldsymbol{\mu}\|_2^2,$$
$$\zeta_{j,r,i}^{(t+1)} = \zeta_{j,r,i}^{(t)} - \frac{\eta}{nm} \cdot \sum_{k \in \mathcal{N}(i)} D_k^{-1} \ell_k'^{(t)} \cdot \sigma'(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_k \rangle) \cdot \mathbb{1}(y_k = j) \|\boldsymbol{\xi}_i\|_2^2.$$

We define $t_{j,r,i}$ as the final moment $t < T^*$ when $\zeta_{j,r,i}^{(t)} \leq 0.5\alpha$.

We can express $\zeta_{j,r,i}^{(\tilde{T})}$ as follows:

$$\zeta_{j,r,i}^{(\tilde{T})} = \zeta_{j,r,i}^{(t_{j,r,i})} \underbrace{- \frac{\eta}{nm} \cdot \sum_{k \in \mathcal{N}(i)} D_k^{-1} \cdot \ell_k'^{(t_{j,r,i})} \cdot \sigma'(\langle \mathbf{w}_{j,r}^{(t_{j,r,i})}, \tilde{\boldsymbol{\xi}}_k \rangle) \cdot \mathbb{1}(y_k = j) \|\boldsymbol{\xi}_i\|_2^2}_{I_1}$$
$$- \underbrace{\sum_{t_{j,r,i} < t < T} \frac{\eta}{nm} \cdot \sum_{k \in \mathcal{N}(i)} D_k^{-1} \cdot \ell_k'^{(t)} \cdot \sigma'(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_k \rangle) \cdot \mathbb{1}(y_k = j) \|\boldsymbol{\xi}_i\|_2^2}_{I_2}. \qquad (23)$$

Next, we aim to establish an upper bound for I_1:

$$|I_1| \leq 2qn^{-1}m^{-1}\eta \bigg(\max_k \hat{\rho}_{j,r,k}^{(t_{j,r,i})} + 0.5\beta + 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha \bigg)^{q-1} \sigma_p^2 d$$
$$\leq q2^q n^{-1}m^{-1}\eta\alpha^{q-1}\sigma_p^2 d \leq 0.25\alpha,$$

where we apply Lemmas B.6 and A.1 for the first inequality, utilize the conditions $\beta \leq 0.1\alpha$ and $8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha \leq 0.1\alpha$ for the second inequality, and finally, the constraint $\eta \leq nm/(q2^{q+2}\alpha^{q-2}\sigma_p^2 d)$ for the last inequality.

Second, we bound I_2. For $t_{j,r,i} < t < \tilde{T}$ and $y_k = j$, we can lower bound $\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_k \rangle$ as follows,

$$\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_k \rangle \geq \langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_k \rangle + \hat{\rho}_{j,r,k}^{(t)} - 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha$$
$$\geq -0.5\beta + \frac{1}{4}\frac{p-s}{p+s}\alpha - 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha$$
$$\geq 0.25\alpha,$$

where the first inequality is by Lemma B.6, the second inequality is by $\hat{\rho}_{j,r,i}^{(t)} > \frac{1}{4}\frac{p-s}{p+s}\alpha$ and $\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i \rangle \geq -0.5\beta$ due to the definition of $t_{j,r,i}$ and β, the last inequality is by $\beta \leq 0.1\alpha$ and $8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha \leq 0.1\alpha$. Similarly, for $t_{j,r,i} < t < \tilde{T}$ and $y_k = j$, we can also upper bound $\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_k \rangle$ as follows,

$$\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_k \rangle \leq \langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_k \rangle + \hat{\rho}_{j,r,k}^{(t)} + 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha$$
$$\leq 0.5\beta + \frac{3}{4}\frac{p-s}{p+s}\alpha + 8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha$$
$$\leq 2\alpha,$$

where the first inequality is by Lemma B.6, the second inequality is by induction hypothesis $\hat{\rho}_{j,r,i}^{(t)} \leq \alpha$, the last inequality is by $\beta \leq 0.1\alpha$ and $8n\sqrt{\frac{\log(4n^2/\delta)}{d}}\alpha \leq 0.1\alpha$.

Hence, we can derive the following expression for I_2:

$$\begin{aligned}
|I_2| &\le \sum_{t_{j,r,i}<t<\tilde{T}} \frac{\eta}{nm} \cdot \sum_{k \in \mathcal{N}(i)} D_k^{-1} \exp(-\sigma(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_k \rangle) + 1) \cdot \sigma'(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_k \rangle) \cdot \mathbb{1}(y_k = j) \|\boldsymbol{\xi}_i\|_2^2 \\
&\le \frac{eq 2^q \eta T^*}{n} \exp(-\alpha^q/4^q) \alpha^{q-1} \sigma_p^2 d \\
&\le 0.25 T^* \exp(-\alpha^q/4^q) \alpha \\
&\le 0.25 T^* \exp(-\log(T^*)^q) \alpha \\
&\le 0.25 \alpha,
\end{aligned}$$

where we apply (22) for the first inequality, utilize Lemma A.1 for the second, employ the constraint $\eta = O\big(nm/(q2^{q+2}\alpha^{q-2}\sigma_p^2 d)\big)$ in (12) for the third, and finally, the conditions $\alpha = 4\log(T^*)$ and $\log(T^*)^q \ge \log(T^*)$ for the subsequent inequalities. By incorporating the bounds of I_1 and I_2 into (23), we conclude the proof for ζ.

In a similar manner, we can establish that $\gamma_{j,r}^{(\tilde{T})} \le \alpha$ by using $\eta = O\big(nm/(q2^{q+2}\alpha^{q-2}\|\boldsymbol{\mu}\|_2^2)\big)$ in (12). Thus, Proposition B.4 is valid for $t = \tilde{T}$, completing the induction process. As a corollary to Proposition B.4, we identify a crucial characteristic of the loss function during training within the interval $0 \le t \le T^*$. This characteristic will play a vital role in the subsequent convergence analysis.

□

C Two Stage Dynamics Analysis

In this section, we employ a two-stage dynamics analysis to investigate the behavior of coefficient iterations. During the first stage, the derivative of the loss function remains almost constant due to the small weight initialization. In the second stage, the derivative of the loss function ceases to be constant, necessitating an analysis that meticulously takes this into account.

C.1 First stage: feature learning versus noise memorization

Lemma C.1 (Restatement of Lemma 5.4)**.** *Under the same conditions as Theorem 4.2, in particular if we choose*

$$n \cdot \mathrm{SNR}^q \cdot (n(p+s))^{q/2-1} \ge C \log(6/\sigma_0 \|\boldsymbol{\mu}\|_2) 2^{2q+6} [4\log(8mn/\delta)]^{(q-1)/2}, \tag{24}$$

where $C = O(1)$ is a positive constant, there exists time $T_1 = \frac{C\log(6/\sigma_0\|\boldsymbol{\mu}\|_2)2^{q+1}m}{\eta\sigma_0^{q-2}\|\boldsymbol{\mu}\|_2^q \Xi^q}$ such that

- $\max_r \gamma_{j,r}^{(T_1)} \ge 2$ *for $j \in \{\pm 1\}$.*
- $|\rho_{j,r,i}^{(t)}| \le \sigma_0 \sigma_p \sqrt{d/(n(p+s))}/2$ *for all $j \in \{\pm1\}, r \in [m]$, $i \in [n]$ and $0 \le t \le T_1$.*

Remark C.2. In this lemma, we establish that the rate of signal learning significantly outpaces that of noise memorization within GNNs. After a specific number of iterations, the GNN is able to learn the signal from the data at a constant or higher order, while only memorizing a smaller order of noise.

Proof of Lemma C.1. Let us define

$$T_1^+ = \frac{nm\eta^{-1}\sigma_0^{2-q}\sigma_p^{-q}d^{-q/2}(n(p+s))^{(q-2)/2}}{2^{q+4}q[4\log(8mn/\delta)]^{(q-2)/2}}. \tag{25}$$

We will begin by establishing the outcome related to noise memorization. Let $\Psi^{(t)}$ be the maximum value over all j, r, i of $|\rho_{j,r,i}^{(t)}|$, that is, $\Psi^{(t)} = \max_{j,r,i}\{\overline{\rho}_{j,r,i}^{(t)}, -\underline{\rho}_{j,r,i}^{(t)}\}$. We will employ an inductive argument to demonstrate that

$$\Psi^{(t)} \le \sigma_0 \sigma_p \sqrt{d/(n(p+s))} \tag{26}$$

is valid for the entire range $0 \leq t \leq T_1^+$. By its very definition, it is evident that $\Psi^{(0)} = 0$. Assuming that there exists a value $\tilde{T} \leq T_1^+$ for which equation (26) is satisfied for all $0 < t \leq \tilde{T} - 1$, we can proceed as follows.

$$
\begin{aligned}
\Psi^{(t+1)} \leq \Psi^{(t)} &+ \frac{\eta}{nm} \sum_{k \in \mathcal{N}(i)} D_k^{-1} \cdot |\ell_k'^{(t)}| \cdot \\
&\sigma'\Bigg(\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_k \rangle + \sum_{i'=1}^{n} \Psi^{(t)} \cdot \frac{|\langle \boldsymbol{\xi}_{i'}, \tilde{\boldsymbol{\xi}}_k \rangle|}{\|\boldsymbol{\xi}_{i'}\|_2^2} + \sum_{i'=1}^{n} \Psi^{(t)} \cdot \frac{|\langle \boldsymbol{\xi}_{i'}, \tilde{\boldsymbol{\xi}}_k \rangle|}{\|\boldsymbol{\xi}_{i'}\|_2^2}\Bigg) \cdot \|\boldsymbol{\xi}_i\|_2^2 \\
&\leq \Psi^{(t)} + \frac{\eta}{nm} \cdot \sum_{k \in \mathcal{N}(i)} D_k^{-1} \sigma'\Bigg(\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_k \rangle + 2 \cdot \sum_{i'=1}^{n} \Psi^{(t)} \cdot \frac{|\langle \boldsymbol{\xi}_{i'}, \tilde{\boldsymbol{\xi}}_k \rangle|}{\|\boldsymbol{\xi}_{i'}\|_2^2}\Bigg) \cdot \|\boldsymbol{\xi}_i\|_2^2 \\
&= \Psi^{(t)} + \frac{\eta}{nm} \cdot \sum_{k \in \mathcal{N}(i)} D_k^{-1} \cdot \\
&\sigma'\Bigg(\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_k \rangle + 2\Psi^{(t)} + 2 \cdot \sum_{i' \neq k'}^{n} \Psi^{(t)} \cdot D_k^{-1} \sum_{k' \in \mathcal{N}(k)} \frac{|\langle \boldsymbol{\xi}_{i'}, \boldsymbol{\xi}_{k'} \rangle|}{\|\boldsymbol{\xi}_{i'}\|_2^2}\Bigg) \cdot \|\boldsymbol{\xi}_i\|_2^2 \\
&\leq \Psi^{(t)} + \frac{\eta q}{nm} \cdot \sum_{k \in \mathcal{N}(i)} D_k^{-1} \Bigg[2 \cdot \sqrt{\log(8mn/\delta)} \cdot \sigma_0 \sigma_p \sqrt{d/(n(p+s))} \\
&+ \Bigg(2 + \frac{4n\sigma_p^2 \cdot \sqrt{d \log(4n^2/\delta)}}{\sigma_p^2 d}\Bigg) \cdot \Psi^{(t)}\Bigg]^{q-1} \cdot 2\sigma_p^2 d \\
&\leq \Psi^{(t)} + \frac{\eta q}{nm} \cdot \big(2 \cdot \sqrt{\log(8mn/\delta)} \cdot \sigma_0 \sigma_p \sqrt{d/(n(p+s))} + 4\Psi^{(t)}\big)^{q-1} \cdot 2\sigma_p^2 d \\
&\leq \Psi^{(t)} + \frac{\eta q}{nm} \cdot \big(4 \cdot \sqrt{\log(8mn/\delta)} \cdot \sigma_0 \sigma_p \sqrt{d/(n(p+s))}\big)^{q-1} \cdot 2\sigma_p^2 d,
\end{aligned}
$$

where the second inequality is due to the constraint $|\ell_i'^{(t)}| \leq 1$, the third inequality is derived from Lemmas A.1 and A.7, the fourth inequality is a consequence of the condition $d \geq 16Dn^2 \log(4n^2/\delta)$, and the final inequality is a result of the inductive assumption (26). Summing over the sequence $t = 0, 1, \ldots, \tilde{T} - 1$, we obtain

$$
\begin{aligned}
\Psi^{(\tilde{T})} &\leq \tilde{T} \cdot \frac{\eta q}{nm} \cdot \big(4 \cdot \sqrt{\log(8mn/\delta)} \cdot \sigma_0 \sigma_p \sqrt{d/(n(p+s))}\big)^{q-1} \cdot 2\sigma_p^2 d \\
&\leq T_1^+ \cdot \frac{\eta q}{nm} \cdot \big(4 \cdot \sqrt{\log(8mn/\delta)} \cdot \sigma_0 \sigma_p \sqrt{d/(n(p+s))}\big)^{q-1} \cdot 2\sigma_p^2 d \\
&\leq \frac{\sigma_0 \sigma_p \sqrt{d/(n(p+s))}}{2},
\end{aligned}
$$

where the second inequality is justified by $\tilde{T} \leq T_1^+$ in our inductive argument. Hence, by induction, we conclude that $\Psi^{(t)} \leq \sigma_0 \sigma_p \sqrt{d/n(p+s)}/2$ for all $t \leq T_1^+$.

Next, we can assume, without loss of generality, that $j = 1$. Let $T_{1,1}$ represent the final time for t within the interval $[0, T_1^+]$ such that $\max_r \gamma_{1,r}^{(t)} \leq 2$, given $\sigma_0 \leq \sqrt{n(p+s)/d}/\sigma_p$. For $t \leq T_{1,1}$, we have $\max_{j,r,i}\{|\rho_{j,r,i}^{(t)}|\} = O(\sigma_0 \sigma_p \sqrt{d/(n(p+s))}) = O(1)$ and $\max_r \gamma_{1,r}^{(t)} \leq 2$. By applying Lemmas B.8 and B.10, we deduce that $F_{-1}(\mathbf{W}_{-1}^{(t)}, \tilde{\mathbf{x}}_i), F_{+1}(\mathbf{W}_{+1}^{(t)}, \tilde{\mathbf{x}}_i) = O(1)$ for all i with $y_i = 1$. Consequently, there exists a positive constant C_1 such that $-\ell_i'^{(t)} \geq C_1$ for all i with $y_i = 1$.

By (7), for $t \leq T_{1,1}$ we have

$$
\begin{aligned}
\gamma_{1,r}^{(t+1)} &= \gamma_{1,r}^{(t)} - \frac{\eta}{nm} \cdot \sum_{i=1}^{n} \ell_i'^{(t)} \cdot \sigma'(\tilde{y}_i \cdot \langle \mathbf{w}_{1,r}^{(0)}, \boldsymbol{\mu} \rangle + \tilde{y}_i \cdot \gamma_{1,r}^{(t)}) \cdot \tilde{y}_i \|\boldsymbol{\mu}\|_2^2 \\
&\geq \gamma_{1,r}^{(t)} + \frac{C_1 \eta}{nm} \cdot \sum_{y_i=1} \sigma'(y_i \Xi \cdot \langle \mathbf{w}_{1,r}^{(0)}, \boldsymbol{\mu} \rangle + y_i \Xi \cdot \gamma_{1,r}^{(t)}) \cdot \frac{p-s}{p+s} \|\boldsymbol{\mu}\|_2^2.
\end{aligned}
$$

Denote $\hat{\gamma}_{1,r}^{(t)} = \gamma_{1,r}^{(t)} + \langle \mathbf{w}_{1,r}^{(0)}, \boldsymbol{\mu} \rangle$ and let $A^{(t)} = \max_r \hat{\gamma}_{1,r}^{(t)}$. Then we have

$$\begin{aligned} A^{(t+1)} &\geq A^{(t)} + \frac{C_1 \eta}{nm} \cdot \sum_{y_i = 1} \sigma'(\Xi A^{(t)}) \cdot \Xi \|\boldsymbol{\mu}\|_2^2 \\ &\geq A^{(t)} + \frac{C_1 \eta q \|\boldsymbol{\mu}\|_2^2}{4m} \Big[\Xi A^{(t)} \Big]^{q-1} \Xi \\ &\geq \bigg(1 + \frac{C_1 \eta q \|\boldsymbol{\mu}\|_2^2}{4m} \big[A^{(0)} \big]^{q-2} \Xi^q \bigg) A^{(t)} \\ &\geq \bigg(1 + \frac{C_1 \eta q \sigma_0^{q-2} \|\boldsymbol{\mu}\|_2^q}{2^q m} \Xi^q \bigg) A^{(t)}, \end{aligned}$$

where the second inequality arises from the lower bound on the quantity of positive data as established in Lemma A.4, the third inequality is a result of the increasing nature of the sequence $A^{(t)}$, and the final inequality is derived from $A^{(0)} = \max_r \langle \mathbf{w}_{1,r}^{(0)}, \boldsymbol{\mu} \rangle \geq \sigma_0 \|\boldsymbol{\mu}\|_2 / 2$, as proven in Lemma A.7. Consequently, the sequence $A^{(t)}$ exhibits exponential growth, and we can express it as

$$\begin{aligned} A^{(t)} &\geq \bigg(1 + \frac{C_1 \eta q \sigma_0^{q-2} \|\boldsymbol{\mu}\|_2^q}{2^q m} \Xi^q \bigg)^t A^{(0)} \\ &\geq \exp\bigg(\frac{C_1 \eta q \sigma_0^{q-2} \|\boldsymbol{\mu}\|_2^q}{2^{q+1} m} \Xi^q t \bigg) A^{(0)} \\ &\geq \exp\bigg(\frac{C_1 \eta q \sigma_0^{q-2} \|\boldsymbol{\mu}\|_2^q}{2^{q+1} m} \Xi^q t \bigg) \frac{\sigma_0 \|\boldsymbol{\mu}\|_2}{2}, \end{aligned}$$

where the second inequality is justified by the relation $1 + z \geq \exp(z/2)$ for $z \leq 2$ and our specific conditions on η and σ_0 as listed in Condition 4.1. The last inequality is a consequence of Lemma A.7 and the definition of $A^{(0)}$. Thus, $A^{(t)}$ will attain the value of 2 within T_1 iterations, defined as

$$T_1 = \frac{\log(6/\sigma_0 \|\boldsymbol{\mu}\|_2) 2^{q+1} m}{C_1 \eta q \sigma_0^{q-2} \|\boldsymbol{\mu}\|_2^q \Xi^q}.$$

Since $\max_r \gamma_{1,r}^{(t)} \geq A^{(t)} - 1$, $\max_r \gamma_{1,r}^{(t)}$ will reach 2 within T_1 iterations. Next, we can confirm that

$$T_1 \leq \frac{nm\eta^{-1} \sigma_0^{2-q} \sigma_p^{-q} d^{-q/2} (n(p+s))^{(q-2)/2}}{2^{q+5} q [4 \log(8mn/\delta)]^{(q-1)/2}} = T_1^+ / 2,$$

where the inequality is consistent with our SNR condition in (24). Therefore, by the definition of $T_{1,1}$, we deduce that $T_{1,1} \leq T_1 \leq T_1^+/2$, utilizing the non-decreasing property of γ. The proof for $j = -1$ follows a similar logic, leading us to the conclusion that $\max_r \gamma_{-1,r}^{(T_{1,-1})} \geq 2$ while $T_{1,-1} \leq T_1 \leq T_1^+/2$, thereby completing the proof.

□

C.2 Second stage: convergence analysis

After the first stage and at time step T_1 we know that:

$$\mathbf{w}_{j,r}^{(T_1)} = \mathbf{w}_{j,r}^{(0)} + j \cdot \gamma_{j,r}^{(T_1)} \cdot \frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|_2^2} + \sum_{i=1}^{n} \zeta_{j,r,i}^{(T_1)} \cdot \frac{\boldsymbol{\xi}_i}{\|\boldsymbol{\xi}_i\|_2^2} + \sum_{i=1}^{n} \omega_{j,r,i}^{(T_1)} \cdot \frac{\boldsymbol{\xi}_i}{\|\boldsymbol{\xi}_i\|_2^2}.$$

And at the beginning of the second stage, we have following property holds:

- $\max_r \gamma_{j,r}^{(T_1)} \geq 2, \forall j \in \{\pm 1\}$.
- $\max_{j,r,i} |\rho_{j,r,i}^{(T_1)}| \leq \hat{\beta}$ where $\hat{\beta} = \sigma_0 \sigma_p \sqrt{d/(n(p+s))}/2$.

Lemma 5.1 implies that the learned feature $\gamma_{j,r}^{(t)}$ will not get worse, i.e., for $t \geq T_1$, we have that $\gamma_{j,r}^{(t+1)} \geq \gamma_{j,r}^{(t)}$, and therefore $\max_r \gamma_{j,r}^{(t)} \geq 2$. Now we choose $\mathbf{W}^*$ as follows:

$$\mathbf{w}_{j,r}^* = \mathbf{w}_{j,r}^{(0)} + 2qm \log(2q/\epsilon) \cdot j \cdot \frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|_2^2}.$$

While the context of CNN presents subtle differences from the scenario described in CNN [8], we can adapt the same analytical approach to derive the following two lemmas:

Lemma C.3 ([8]). *Under the same conditions as Theorem 4.2, we have that* $\|\mathbf{W}^{(T_1)} - \mathbf{W}^*\|_F \leq \tilde{O}(m^{3/2}\|\boldsymbol{\mu}\|_2^{-1})$.

Lemma C.4 ([8]). *Under the same conditions as Theorem 4.2, we have that*

$$\|\mathbf{W}^{(t)} - \mathbf{W}^*\|_F^2 - \|\mathbf{W}^{(t+1)} - \mathbf{W}^*\|_F^2 \geq (2q-1)\eta L_{\mathcal{S}}(\mathbf{W}^{(t)}) - \eta\epsilon$$

for all $T_1 \leq t \leq T^*$.

Lemma C.5 (Restatement of Lemma 5.5). *Under the same conditions as Theorem 4.2, let* $T = T_1 + \left\lfloor \frac{\|\mathbf{W}^{(T_1)} - \mathbf{W}^*\|_F^2}{2\eta\epsilon} \right\rfloor = T_1 + \tilde{O}(m^3\eta^{-1}\epsilon^{-1}\|\boldsymbol{\mu}\|_2^{-2})$. *Then we have* $\max_{j,r,i} |\rho_{j,r,i}^{(t)}| \leq 2\hat{\beta} = \sigma_0\sigma_p\sqrt{d/(n(p+s))}$ *for all* $T_1 \leq t \leq T$. *Besides,*

$$\frac{1}{t - T_1 + 1}\sum_{s=T_1}^{t} L_{\mathcal{S}}(\mathbf{W}^{(s)}) \leq \frac{\|\mathbf{W}^{(T_1)} - \mathbf{W}^*\|_F^2}{(2q-1)\eta(t - T_1 + 1)} + \frac{\epsilon}{2q-1}$$

for all $T_1 \leq t \leq T$*, and we can find an iterate with training loss smaller than* ϵ *within* T *iterations.*

Proof of Lemma C.5. We adapt the convergence proof for CNN[8] to extend the analysis to GNN. By invoking Lemma C.4, for any given time interval $t \in [T_1, T]$, we can deduce that

$$\|\mathbf{W}^{(s)} - \mathbf{W}^*\|_F^2 - \|\mathbf{W}^{(s+1)} - \mathbf{W}^*\|_F^2 \geq (2q-1)\eta L_{\mathcal{S}}(\mathbf{W}^{(s)}) - \eta\epsilon,$$

which is valid for $s \leq t$. Summing over this interval, we arrive at

$$\sum_{s=T_1}^{t} L_{\mathcal{S}}(\mathbf{W}^{(s)}) \leq \frac{\|\mathbf{W}^{(T_1)} - \mathbf{W}^*\|_F^2 + \eta\epsilon(t - T_1 + 1)}{(2q-1)\eta}. \tag{27}$$

This inequality holds for all $T_1 \leq t \leq T$. Dividing both sides of (27) by $(t - T_1 + 1)$, we obtain

$$\frac{1}{t - T_1 + 1}\sum_{s=T_1}^{t} L_{\mathcal{S}}(\mathbf{W}^{(s)}) \leq \frac{\|\mathbf{W}^{(T_1)} - \mathbf{W}^*\|_F^2}{(2q-1)\eta(t - T_1 + 1)} + \frac{\epsilon}{2q-1}.$$

By setting $t = T$, we find that

$$\frac{1}{T - T_1 + 1}\sum_{s=T_1}^{T} L_{\mathcal{S}}(\mathbf{W}^{(s)}) \leq \frac{\|\mathbf{W}^{(T_1)} - \mathbf{W}^*\|_F^2}{(2q-1)\eta(T - T_1 + 1)} + \frac{\epsilon}{2q-1} \leq \frac{3\epsilon}{2q-1} < \epsilon,$$

where we utilize the condition that $q > 2$ and the specific choice of $T = T_1 + \left\lfloor \frac{\|\mathbf{W}^{(T_1)} - \mathbf{W}^*\|_F^2}{2\eta\epsilon} \right\rfloor$. Since the mean value is less than ϵ, it follows that there must exist a time interval $T_1 \leq t \leq T$ for which $L_{\mathcal{S}}(\mathbf{W}^{(t)}) < \epsilon$.

Finally, we aim to demonstrate that $\max_{j,r,i} |\rho_{j,r,i}^{(t)}| \leq 2\hat{\beta}$ holds for all $t \in [T_1, T]$. By inserting $T = T_1 + \left\lfloor \frac{\|\mathbf{W}^{(T_1)} - \mathbf{W}^*\|_F^2}{2\eta\epsilon} \right\rfloor$ into equation (27), we obtain

$$\sum_{s=T_1}^{T} L_{\mathcal{S}}(\mathbf{W}^{(s)}) \leq \frac{2\|\mathbf{W}^{(T_1)} - \mathbf{W}^*\|_F^2}{(2q-1)\eta} = \tilde{O}(\eta^{-1}m^3\|\boldsymbol{\mu}\|_2^2), \tag{28}$$

where the inequality is a consequence of $\|\mathbf{W}^{(T_1)} - \mathbf{W}^*\|_F \leq \tilde{O}(m^{3/2}\|\boldsymbol{\mu}\|_2^{-1})$ as shown in Lemma C.3.

Let's define $\Psi^{(t)} = \max_{j,r,i} |\rho_{j,r,i}^{(t)}|$. We will employ induction to prove $\Psi^{(t)} \leq 2\hat{\beta}$ for all $t \in [T_1, T]$. At $t = T_1$, by the definition of $\hat{\beta}$, it is clear that $\Psi^{(T_1)} \leq \hat{\beta} \leq 2\hat{\beta}$.

Assuming that there exists $\tilde{T} \in [T_1, T]$ such that $\Psi^{(t)} \leq 2\hat{\beta}$ for all $t \in [T_1, \tilde{T} - 1]$, we can consider $t \in [T_1, \tilde{T} - 1]$. Using the expression:

$$\begin{aligned}\rho_{j,r,i}^{(t+1)} = \rho_{j,r,i}^{(t)} - \frac{\eta}{nm} \cdot \sum_{k \in \mathcal{N}(i)} D_k^{-1} \ell_k'^{(t)}\\ \sigma'\Bigg(\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_k \rangle + \sum_{i'=1}^{n} \zeta_{j,r,i'}^{(t)} \frac{\langle \boldsymbol{\xi}_{i'}, \tilde{\boldsymbol{\xi}}_k \rangle}{\|\boldsymbol{\xi}_{i'}\|_2^2} + \sum_{i'=1}^{n} \omega_{j,r,i'}^{(t)} \frac{\langle \boldsymbol{\xi}_{i'}, \tilde{\boldsymbol{\xi}}_k \rangle}{\|\boldsymbol{\xi}_{i'}\|_2^2}\Bigg) \cdot \|\boldsymbol{\xi}_i\|_2^2\end{aligned} \tag{29}$$

we can proceed to analyze:

$$\begin{aligned}\Psi^{(t+1)} &\leq \Psi^{(t)} + \max_{j,r,i}\Bigg\{\frac{\eta}{nm} \cdot \sum_{k \in \mathcal{N}(i)} D_k^{-1} |\ell_k'^{(t)}| \cdot \sigma'\Bigg(\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_k \rangle + 2\sum_{i'=1}^{n} \Psi^{(t)} \cdot \frac{|\langle \boldsymbol{\xi}_{i'}, \tilde{\boldsymbol{\xi}}_k \rangle|}{\|\boldsymbol{\xi}_{i'}\|_2^2}\Bigg) \cdot \|\boldsymbol{\xi}_i\|_2^2\Bigg\}\\
&= \Psi^{(t)} + \max_{j,r,i}\Bigg\{\frac{\eta}{nm} \cdot \sum_{k \in \mathcal{N}(i)} D_k^{-1} |\ell_k'^{(t)}| \cdot\\
&\quad \sigma'\Bigg(\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_k \rangle + 2\Psi^{(t)} + 2\sum_{i' \neq k'}^{n} \Psi^{(t)} \cdot D_k^{-1} \sum_{k' \in \mathcal{N}(k)} \frac{|\langle \boldsymbol{\xi}_{i'}, \boldsymbol{\xi}_{k'} \rangle|}{\|\boldsymbol{\xi}_{i'}\|_2^2}\Bigg) \cdot \|\boldsymbol{\xi}_i\|_2^2\Bigg\}\\
&\leq \Psi^{(t)} + \frac{\eta q}{nm} \cdot \max_i \sum_{k \in \mathcal{N}(i)} D_k^{-1} |\ell_k'^{(t)}| \cdot \Bigg[2 \cdot \sqrt{\log(8mn/\delta)} \cdot \sigma_0 \sigma_p \sqrt{d/(n(p+s))}\\
&\quad + \Bigg(2 + \frac{4n\sigma_p^2 \cdot \sqrt{d \log(4n^2/\delta)}}{\sigma_p^2 d/2}\Bigg) \cdot \Psi^{(t)}\Bigg]^{q-1} \cdot 2\sigma_p^2 d\\
&\leq \Psi^{(t)} + \frac{\eta q}{nm} \cdot \max_i \sum_{k \in \mathcal{N}(i)} D_k^{-1} |\ell_k'^{(t)}| \cdot\\
&\quad \big(2 \cdot \sqrt{\log(8mn/\delta)} \cdot \sigma_0 \sigma_p \sqrt{d/(n(p+s))} + 4 \cdot \Psi^{(t)}\big)^{q-1} \cdot 2\sigma_p^2 d.\end{aligned}$$

The second inequality is derived from Lemmas A.1 and A.7, while the final inequality is based on the assumption that $d \geq 16n^2 \log(4n^2/\delta)$. By taking a telescoping sum, we can express the following:

$$\begin{aligned}\Psi^{(T)} &\overset{(i)}{\leq} \Psi^{(T_1)} + \frac{\eta q}{nm} \sum_{s=T_1}^{\tilde{T}-1} \max_i \sum_{k \in \mathcal{N}(i)} D_k^{-1} |\ell_k'^{(t)}| \tilde{O}(\sigma_p^2 d) \hat{\beta}^{q-1}\\
&\overset{(ii)}{\leq} \Psi^{(T_1)} + \frac{\eta q}{nm} \tilde{O}(\sigma_p^2 d) \hat{\beta}^{q-1} \sum_{s=T_1}^{\tilde{T}-1} \max_i \sum_{k \in \mathcal{N}(i)} D_k^{-1} \ell_k^{(s)}\\
&\overset{(iii)}{\leq} \Psi^{(T_1)} + \tilde{O}(\eta m^{-1} \sigma_p^2 d) \hat{\beta}^{q-1} \sum_{s=T_1}^{\tilde{T}-1} L_{\mathcal{S}}(\mathbf{W}^{(s)})\\
&\overset{(iv)}{\leq} \Psi^{(T_1)} + \tilde{O}(m^2 \mathrm{SNR}^{-2}) \hat{\beta}^{q-1}\\
&\overset{(v)}{\leq} \hat{\beta} + \tilde{O}(m^2 n^{2/q} (n(p+s))^{1-2/q} \hat{\beta}^{q-2}) \hat{\beta}\\
&\overset{(vi)}{\leq} 2\hat{\beta},\end{aligned}$$

where (i) follows from our induction assumption that $\Psi^{(t)} \leq 2\hat{\beta}$, (ii) is derived from the relationship $|\ell'| \leq \ell$, (iii) is obtained by the sum of $\max_i \sum_{k \in \mathcal{N}(i)} D_k^{-1} \leq \sum_i \ell_i^{(s)} = nL_{\mathcal{S}}(\mathbf{W}^{(s)})$, (iv) is due to the summation of $\sum_{s=T_1}^{\tilde{T}-1} L_S(\mathbf{W}^{(s)}) \leq \sum_{s=T_1}^{T} L_{\mathcal{S}}(\mathbf{W}^{(s)}) = \tilde{O}(\eta^{-1} m^3 \|\boldsymbol{\mu}\|_2^2)$ as shown in (28), (v) is based on the condition $n\mathrm{SNR}^q \cdot$

$(n(p+s))^{q/2-1} \geq \tilde{\Omega}(1)$, and (vi) follows from the definition of $\hat{\beta} = \sigma_0\sigma_p\sqrt{d/(n(p+s))}/2$ and $\tilde{O}(m^2 n^{2/q}(n(p+s))^{1-2/q}\hat{\beta}^{q-2}) = \tilde{O}(m^2 n^{2/q}(n(p+s))^{1-2/q}(\sigma_0\sigma_p\sqrt{d/(n(p+s))})^{q-2}) \leq 1$.

Thus, we conclude that $\Psi^{(\tilde{T})} \leq 2\hat{\beta}$, completing the induction and establishing the desired result. □

D Population loss

Consider a new data point $(\mathbf{x}, y)$ drawn from the SNM-SBM distribution. Without loss of generality, we suppose that the first patch is the signal patch and the second patch is the noise patch, i.e., $\mathbf{x} = [y \cdot \boldsymbol{\mu}, \boldsymbol{\xi}]$. Moreover, by the signal-noise decomposition, the learned neural network has parameter:

$$\mathbf{w}_{j,r}^{(t)} = \mathbf{w}_{j,r}^{(0)} + j \cdot \gamma_{j,r}^{(t)} \cdot \frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|_2^2} + \sum_{i=1}^{n} \zeta_{j,r,i}^{(t)} \cdot \frac{\boldsymbol{\xi}_i}{\|\boldsymbol{\xi}_i\|_2^2} + \sum_{i=1}^{n} \omega_{j,r,i}^{(t)} \cdot \frac{\boldsymbol{\xi}_i}{\|\boldsymbol{\xi}_i\|_2^2}$$

for $j \in \{\pm 1\}$ and $r \in [m]$.

Although the framework of MLP diverges in certain nuances from the situation of MLP outlined in [8], we are able to employ a similar analytical methodology to deduce the subsequent two lemmas:

Lemma D.1. *Under the same conditions as Theorem 4.2, we have that* $\max_{j,r} |\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_i \rangle| \leq 1/2$ *for all* $0 \leq t \leq T$, *and* $i \in [n]$.

Lemma D.2. *Under the same conditions as Theorem 4.2, with probability at least* $1 - 4mT \cdot \exp(-C_2^{-1}\sigma_0^{-2}\sigma_p^{-2}d^{-1}n(p+s))$, *we have that* $\max_{j,r} |\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}} \rangle| \leq 1/2$ *for all* $0 \leq t \leq T$, *where* $C_2 = \tilde{O}(1)$.

Lemma D.3 (Restatement of Lemma 5.6)**.** *Let T be defined in Lemma 5.4 respectively. Under the same conditions as Theorem 4.2, for any* $0 \leq t \leq T$ *with* $L_S(\mathbf{W}^{(t)}) \leq 1$, *it holds that* $L_{\mathcal{D}}(\mathbf{W}^{(t)}) \leq c_1 \cdot L_S(\mathbf{W}^{(t)}) + \exp(-c_2 n^2)$.

Proof of Lemma D.3. Consider the occurrence of event $\mathcal{E}$, defined as the condition under which Lemma D.2 is satisfied. We can then express the loss $L_{\mathcal{D}}(\mathbf{W}^{(t)})$ as a sum of two components:

$$\mathbb{E}\big[\ell\big(yf(\mathbf{W}^{(t)}, \tilde{\mathbf{x}})\big)\big] = \underbrace{\mathbb{E}[\mathbb{1}(\mathcal{E})\ell\big(yf(\mathbf{W}^{(t)}, \tilde{\mathbf{x}})\big)]}_{\text{Term } I_1} + \underbrace{\mathbb{E}[\mathbb{1}(\mathcal{E}^c)\ell\big(yf(\mathbf{W}^{(t)}, \tilde{\mathbf{x}})\big)]}_{\text{Term } I_2}. \tag{30}$$

Next, we proceed to establish bounds for I_1 and I_2.

Bounding I_1: Given that $L_{\mathcal{S}}(\mathbf{W}^{(t)}) \leq 1$, there must be an instance $(\tilde{\mathbf{x}}_i, y_i)$ for which $\ell\big(y_i f(\mathbf{W}^{(t)}, \tilde{\mathbf{x}}_i)\big) \leq L_{\mathcal{S}}(\mathbf{W}^{(t)}) \leq 1$, leading to $y_i f(\mathbf{W}^{(t)}, \tilde{\mathbf{x}}_i) \geq 0$. Hence, we obtain:

$$\exp(-y_i f(\mathbf{W}^{(t)}, \tilde{\mathbf{x}}_i)) \overset{(i)}{\leq} 2\log\big(1 + \exp(-y_i f(\mathbf{W}^{(t)}, \tilde{\mathbf{x}}_i))\big) = 2\ell\big(y_i f(\mathbf{W}^{(t)}, \tilde{\mathbf{x}}_i)\big) \leq 2L_{\mathcal{S}}(\mathbf{W}^{(t)}), \tag{31}$$

where (i) follows from the inequality $z \leq 2\log(1+z), \forall z \leq 1$. If event $\mathcal{E}$ occurs, we deduce:

$$\begin{aligned} |yf(\mathbf{W}^{(t)}, \tilde{\mathbf{x}}^{(2)}) - y_i f(\mathbf{W}^{(t)}, \tilde{\mathbf{x}}_i^{(2)})| &\leq \frac{1}{m}\sum_{j,r} \sigma(\langle \mathbf{w}_{j,r}, \tilde{\boldsymbol{\xi}}_i \rangle) + \frac{1}{m}\sum_{j,r} \sigma(\langle \mathbf{w}_{j,r}, \tilde{\boldsymbol{\xi}} \rangle) \\ &\leq 1. \end{aligned} \tag{32}$$

Here, $f(\mathbf{W}^{(t)}, \tilde{\mathbf{x}}^{(2)})$ refers to the input $\tilde{\mathbf{x}} = [0, \tilde{\mathbf{x}}^{(2)}]$. The second inequality is justified by Lemmas D.2 and D.1. Consequently, we have:

$$
\begin{aligned}
I_1 &\leq \mathbb{E}[\mathbb{1}(\mathcal{E})\exp(-yf(\mathbf{W}^{(t)},\tilde{\mathbf{x}}))] \\
&= \mathbb{E}[\mathbb{1}(\mathcal{E})\exp(-y_i f(\mathbf{W}^{(t)},\tilde{\mathbf{x}}^{(1)}))\exp(-y_i f(\mathbf{W}^{(t)},\tilde{\mathbf{x}}^{(2)}))] \\
&\leq 2e\cdot C\cdot \mathbb{E}[\mathbb{1}(\mathcal{E})\exp(-y_i f(\mathbf{W}^{(t)},\tilde{\mathbf{x}}_i^{(1)}))\exp(-y_i f(\mathbf{W}^{(t)},\tilde{\mathbf{x}}_i^{(2)}))] \\
&\leq 2e\cdot \mathbb{E}[\mathbb{1}(\mathcal{E})L_{\mathcal{S}}(\mathbf{W}^{(t)})],
\end{aligned}
$$

where the inequalities follow from the properties of cross-entropy loss, (32), Lemma A.4, and (31). The constant c_1 encapsulates the factors in the derivation.

Estimating I_2: We now turn our attention to the second term I_2. By selecting an arbitrary training data point $(\mathbf{x}_{i'}, y_{i'})$ with $y_{i'} = y$, we can derive the following:

$$
\begin{aligned}
\ell\big(yf(\mathbf{W}^{(t)},\tilde{\mathbf{x}})\big) &\leq \log(1+\exp(F_{-y}(\mathbf{W}^{(t)},\tilde{\mathbf{x}}))) \\
&\leq 1 + F_{-y}(\mathbf{W}^{(t)},\tilde{\mathbf{x}}) \\
&= 1 + \frac{1}{m}\sum_{j=-y,r\in[m]}\sigma(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{y}\boldsymbol{\mu}\rangle) + \frac{1}{m}\sum_{j=-y,r\in[m]}\sigma(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}\rangle) \\
&\leq 1 + F_{-y_i}(\mathbf{W}_{-y_{i'}}, \tilde{\mathbf{x}}_{i'}) + \frac{1}{m}\sum_{j=-y,r\in[m]}\sigma(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}\rangle) \\
&\leq 2 + \frac{1}{m}\sum_{j=-y,r\in[m]}\sigma(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}\rangle) \\
&\leq 2 + \tilde{O}((\sigma_0\sqrt{d})^q)\|\tilde{\boldsymbol{\xi}}\|^q,
\end{aligned} \tag{33}
$$

where the inequalities follow from the properties of the cross-entropy loss and the constraints defined in Lemma B.8. The last inequality is a result of the boundedness of the inner product with $\tilde{\boldsymbol{\xi}}$. Continuing, we have:

$$
\begin{aligned}
I_2 &\leq \sqrt{\mathbb{E}[\mathbb{1}(\mathcal{E}^c)]}\cdot\sqrt{\mathbb{E}\Big[\ell\big(yf(\mathbf{W}^{(t)},\tilde{\mathbf{x}})\big)^2\Big]} \\
&\leq \sqrt{\mathbb{P}(\mathcal{E}^c)}\cdot\sqrt{4+\tilde{O}((\sigma_0\sqrt{d})^{2q})\mathbb{E}[\|\tilde{\boldsymbol{\xi}}\|_2^{2q}]} \\
&\leq \exp\left[-\tilde{\Omega}\left(\frac{\sigma_0^{-2}\sigma_p^{-2}}{d^{-1}n(p+s)}\right) + \mathrm{polylog}(d)\right] \\
&\leq \exp(-c_1 n^2),
\end{aligned}
$$

where c_1 is a constant, the first inequality is by Cauchy-Schwartz inequality, the second inequality is by (33), the third inequality is by Lemma D.2 and the fact that $\sqrt{4+\tilde{O}((\sigma_0\sqrt{d})^{2q})\mathbb{E}[\|\tilde{\boldsymbol{\xi}}\|_2^{2q}]} = O(\mathrm{poly}(d))$, and the last inequality is by our condition $\sigma_0 \leq \tilde{O}(m^{-2/(q-2)}n^{-1})\cdot(\sigma_p\sqrt{d/(n(p+s))})^{-1}$ in Condition 4.1. Plugging the bounds of I_1, I_2 completes the proof. □

E Parallels between our data model and real-world dataset

The citation network (Cora, Citeseer, and Pubmed) employ a bag-of-words feature representation, typically represented by one-hot vectors, thereby ensuring orthogonality between features. We can conceptually divide words into two categories: label-relevant and label-irrelevant. For example, words like "algorithm" or "neural network" are label-relevant to the subject of computer science, while general words like "study" or "approach" are label-irrelevant. In our SNM, $\boldsymbol{\mu}$ represents label-relevant features, while $\boldsymbol{\xi}$ represents label-irrelevant ones.

Furthermore, the datasets Wiki-CS, Amazon-Computers, Amazon-Photo, Coauthor-CS, and Coauthor-Physics [48] also parallels with our theoretical model and we provide the more discussion as follows:

- The Cora dataset includes 2,708 scientific publications, each categorized into one of seven classes, connected by 5,429 links. Each publication is represented by a binary word vector, which denotes the presence or absence of a corresponding word from a dictionary of 1,433 unique words.
- The Citeseer dataset comprises 3,312 scientific publications, each classified into one of six classes, connected by 4,732 links. Each publication is represented by a binary word vector, indicating the presence or absence of a corresponding word from a dictionary that includes 3,703 unique words.
- The Pubmed Diabetes dataset includes 19,717 scientific publications related to diabetes, drawn from the PubMed database and classified into one of three classes. The citation network is made up of 44,338 links. Each publication is represented by a TF-IDF weighted word vector from a dictionary consisting of 500 unique words.
- Coauthor CS (Computer Science) & Coauthor Physics (Coauthor Phy.): The dataset typically includes features based on the keywords of an author's papers, and the task is often to predict each author's research field or interests based on their publication record and collaboration network.
- Amazon Computers & Amazon Photo: Node features are derived from product reviews, and the classification task involves predicting product categories based on the co-purchase relationships and review data.
- WikiCS Node features could be derived from the text of the articles, such as word vectors. The classification task usually involves categorizing articles into different areas or subjects within Computer Science based on their content and the article network structure.

We have broadened our analysis to include the measurement of cosine similarity between two equal-sized parts of node features (excluding the final feature for odd-sized representations) across a diverse range of datasets. This extended analysis bolsters the orthogonality relation posited in our model. The results are presented in Table 1.

Dataset	**Feature Dimension**	**Cossin Similarity**
Cora	1433	1.57×10^{-5}
Citeseer	3703	3.99×10^{-6}
Pubmed	500	2.00×10^{-4}
Coauthor CS	6805	2.28×10^{-6}
Coauthor Phy.	8451	1.08×10^{-6}
Amazon Comp.	767	9.00×10^{-4}
Amazon Photo	745	9.00×10^{-4}
WikiCS	300	1.00×10^{-4}

Table 1: Cosine similarity analysis of node features across various datasets.

F Phase transition in GCN

In Figure 3, we illustrated the variance in test accuracy between MLP and GCN within a chosen range of SNR and sample numbers, where GCN was shown to achieve near-perfect test accuracy. Here, we broaden the SNR range towards the smaller end and display the corresponding phase diagram of GCN in Figure 8. When the SNR is exceedingly small, we observe that GCNs return lower test accuracy, suggesting the possibility of a phase transition in the test accuracy of GCNs.

G How to Transform MNIST into a Signal-Noise Data Model

To verify the theoretical study, we used the real-world data MNIST and modified it to align with the theoretical data model. In particular, we added Gaussian noise to the position of the image border while retaining the numbers in the middle. The final renderings are shown in Figure 9, where the noise level is chosen as $\sigma_p = 0.5$.

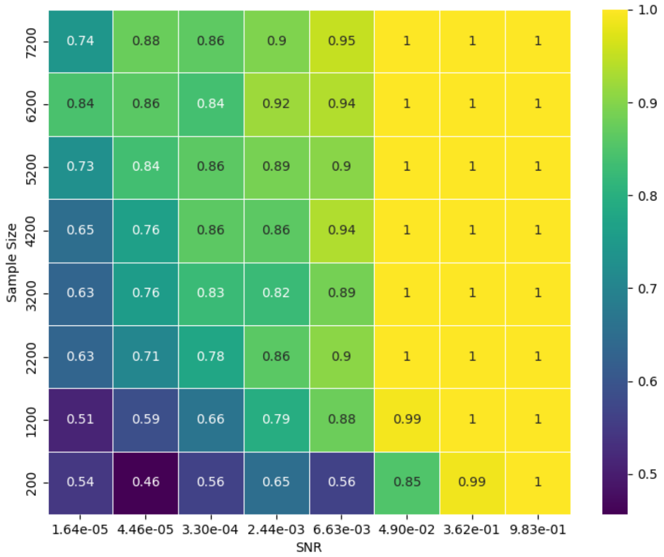


Figure 8: Test accuracy heatmap for GCNs after training.

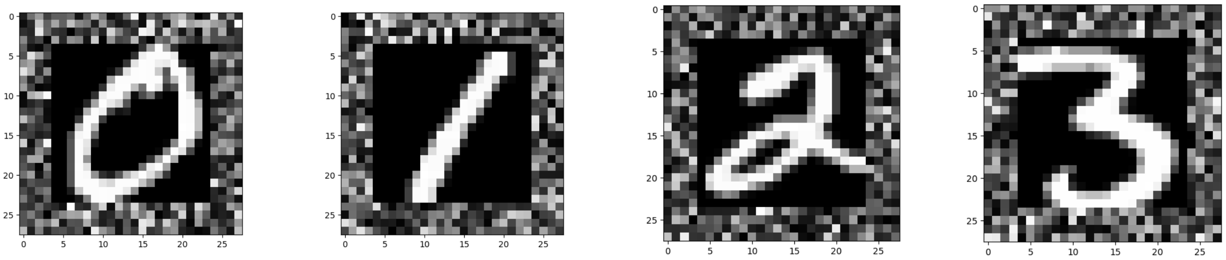


Figure 9: Examples of adding noise patch to the numbers in the MNIST dataset.

A similar strategy can be found in [8]. As can be seen from Figure 9, the surrounding noise forms a patch, and the numbers in the middle form a signal patch. In subsequent experiments, we will separate noise patches and signal patches.

H Computation Resources

We implement our methods with PyTorch. For the software and hardware configurations, we ensure the consistent environments for each datasets. We run all the experiments on Linux servers with NVIDIA V100 graphics cards with CUDA 11.2.

I Broader Impacts

This work focuses on the theoretical understanding of the differences in optimization and generalization between GNNs and MLPs. The results established for GNNs may be applied to both theoretical and empirical research on GNNs. Additionally, we do not foresee any form of negative social impact induced by our work.