Dynamics of Nonlinear Feature Learning in Two-Layer GCNs on XOR-CSBM

author names withheld

Under Review for the Workshop on High-dimensional Learning Dynamics, 2026

Abstract

Graph Convolutional Networks (GCNs) are widely used for graph-structured data, but their training dynamics are not well understood. We study nonlinear feature learning in a two-layer GCN through a minimal XOR Contextual Stochastic Block Model (XOR-CSBM), where successful prediction requires nonlinear feature formation. By analyzing the population-gradient dynamics, we derive a tractable difference equation for the relevant representation variables. This reveals that feature learning proceeds through two distinct phases and ultimately yields task-relevant nonlinear representations.

1. Introduction

Graph Neural Networks (GNNs) have achieved strong empirical success on graph-structured data in applications such as molecular modeling, recommendation systems, and traffic forecasting ([5], [15], [16]). Among them, Graph Convolutional Networks (GCNs) are particularly attractive due to their simplicity and effectiveness. However, the mechanisms underlying GCN training remain incompletely understood ([2], [8]). In particular, it remains unclear how GCNs realize nonlinear feature learning in the presence of graph aggregation.

To study this question, we consider a minimal XOR-CSBM learning problem, where successful prediction necessarily requires nonlinear feature formation from graph-structured data ([4]). We analyze a two-layer nonlinear GCN trained by online gradient descent and derive a tractable population-level difference equation for the relevant representation variables. This analysis shows that the GCN learns nonlinear features through two phases of training dynamics.

Our contributions are as follows. We formulate a minimal XOR-CSBM setting for nonlinear feature learning in a two-layer nonlinear GCN, derive a population-level difference equation characterizing its training dynamics, and show that feature learning proceeds through two phases that ultimately yield task-relevant nonlinear representations.

1.1. Related Work

Graph Convolutional Networks (GCNs). GCNs, introduced by [10], are a fundamental GNN architecture for graph-structured data. Their theoretical properties have been studied from several perspectives. For example, [3] and [9] established generalization error bounds, while [14] analyzed oversmoothing in CSBMs.

CSBM and XOR-CSBM. The Contextual Stochastic Block Model (CSBM) is a widely used generative model for theoretical analysis. In this framework, [13] studied semi-supervised learning

for GCNs. The XOR model is a minimal linearly inseparable feature model, and [6] analyzed SGD dynamics for neural networks on XOR. The XOR-CSBM combines CSBM with XOR-type feature distributions, making the classification task nonlinear and more challenging than in standard CSBMs. [4] analyzed XOR-CSBM to clarify the role of graph convolutional layers.

Gradient descent for GCNs. Understanding gradient-based learning dynamics is important for explaining the behavior of neural networks. While online gradient descent was first analyzed for standard neural networks ([12]), later work extended such analyses to GNNs. For example, [11] developed an NTK framework for infinite-width GNNs, [2] analyzed gradient descent for two-layer and linear multi-layer GCNs, and [8] showed that graph structure can make GCN optimization more intricate than that of MLPs by reducing feature noise.

1.2. Notation

We use boldface letters to denote vectors. For a vector $\boldsymbol{x}$, $\|\boldsymbol{x}\|$ denotes its Euclidean norm. For an integer n, we write $[n] = \{1, \dots, n\}$. $\boldsymbol{I}_d$ denotes the $d \times d$ identity matrix. The ReLU function is defined by $\mathrm{ReLU}(x) = \max(0, x)$. For a vector $\boldsymbol{x}$, $\mathrm{ReLU}(\boldsymbol{x})$ denotes the element-wise application of the ReLU function. $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ denotes a Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix Σ. $\mathrm{Ber}(p)$ denotes a Bernoulli distribution. For a set A, $\mathrm{Unif}(A)$ denotes a uniform distribution on A.

2. Setup

2.1. XOR-CSBM

We use the XOR-CSBM model to generate the data, which is a combination of an XOR-Gaussian Mixture Model for feature generation and a Stochastic Block Model for graph structure generation.

Definition 1 *(XOR-Gaussian Mixture Model, XOR-GMM) Let $\boldsymbol{\mu}_1, \boldsymbol{\mu}_{-1} \in \mathbb{R}^d$ be two orthonormal vectors, i.e., $\|\boldsymbol{\mu}_1\| = \|\boldsymbol{\mu}_{-1}\| = 1$ and $\boldsymbol{\mu}_1^\top \boldsymbol{\mu}_{-1} = 0$. First, sample a label $y_i \sim \mathrm{Unif}(\{\pm 1\})$. Then, sample N independent feature vectors $\{\boldsymbol{x}_i | i \in [N]\}$ as follows:*

$$\boldsymbol{x}_i \sim \begin{cases} \frac{1}{2}\mathcal{N}(\theta\boldsymbol{\mu}_1, I_d) + \frac{1}{2}\mathcal{N}(-\theta\boldsymbol{\mu}_1, I_d), & y_i = 1, \\ \frac{1}{2}\mathcal{N}(\theta\boldsymbol{\mu}_{-1}, I_d) + \frac{1}{2}\mathcal{N}(-\theta\boldsymbol{\mu}_{-1}, I_d), & y_i = -1 \end{cases}$$

where $\theta > 0$ controls the signal-to-noise ratio (SNR). In addition, we define $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N]^\top \in \mathbb{R}^{N \times d}$, $\boldsymbol{y} = (y_1 \cdots y_N)^\top \in \{\pm 1\}^N$ and write $\boldsymbol{X} \sim XOR(\boldsymbol{\mu}_1, \boldsymbol{\mu}_{-1}, \boldsymbol{y}, \theta)$ if $\{(\boldsymbol{x}_i, y_i) |\ i \in [N]\}$ is generated by the above rules.

We use the Stochastic Block Model (SBM) [7] to generate the graph structure. SBM is a random graph model that generates intra-cluster edges with probability α, inter-cluster edges with probability β and no self-loops.

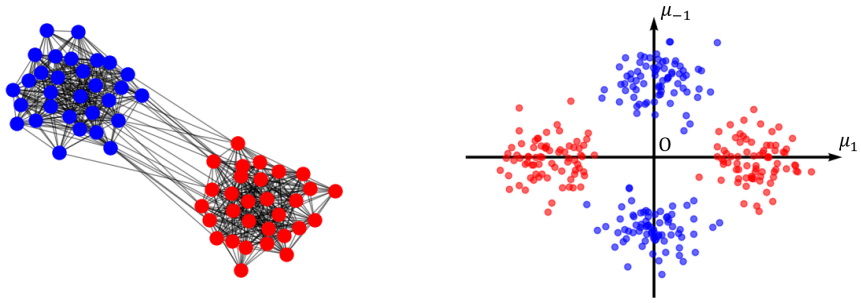
Definition 2 *(Stochastic Block Model, SBM) Let $0 < \alpha, \beta < 1$ be the probabilities of the intra-cluster and inter-cluster edges. We denote its adjacency matrix as $\boldsymbol{A} = (A_{ij})_{i,j=1,\cdots,N}$, and the entry is generated as follows:*

$$A_{ii} = 0, \quad A_{ij} \sim \begin{cases} \mathrm{Ber}(\alpha) & \text{if } i < j,\ y_i = y_j \\ \mathrm{Ber}(\beta) & \text{if } i < j,\ y_i \neq y_j \end{cases}, \quad A_{ij} = A_{ji} \text{ for } i > j$$

Note that the generation of the graph structure is independent of $\boldsymbol{X}$ conditionally on $\boldsymbol{y}$. Analogously to the XOR model, we write $\boldsymbol{A} \sim \mathrm{SBM}(\boldsymbol{y}, \alpha, \beta)$.

This paper focuses on the XOR-CSBM, which is the rule of generating features and the graph structure as follows:

Definition 3 *(XOR-Contextual Stochastic Block Model, XOR-CSBM) Suppose that* $N, d \in \mathbb{N}_+$, $0 \leq \alpha, \beta \leq 1$, $\theta > 0$, $\boldsymbol{\mu}_1, \boldsymbol{\mu}_{-1} \in \mathbb{R}^d$ *satisfy* $\|\boldsymbol{\mu}_1\| = \|\boldsymbol{\mu}_{-1}\| = 1$ *and* $\boldsymbol{\mu}_1^T \boldsymbol{\mu}_{-1} = 0$. *We write* $(\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{A}) \sim \mathrm{XOR\text{-}CSBM}(\boldsymbol{\mu}_1, \boldsymbol{\mu}_{-1}, \alpha, \beta, \theta)$ *when* $\boldsymbol{X}$ *and* $\boldsymbol{A}$ *are generated by (1)* $\boldsymbol{y} \sim \mathrm{Unif}(\{\pm 1\}^N)$, *(2) given* $\boldsymbol{y}$, θ *and* $\boldsymbol{\mu}_{\pm 1}$, $\boldsymbol{X} \sim XOR(\boldsymbol{\mu}_1, \boldsymbol{\mu}_{-1}, \boldsymbol{y}, \theta)$, *(3) given* $\boldsymbol{y}$, α *and* β, $\boldsymbol{A} \sim \mathrm{SBM}(\boldsymbol{y}, \alpha, \beta)$.



(a) The example of the graph generated by SBM

(b) The example of features generated by XOR-GMM

Figure 1: The examples of the graph generated by SBM and features generated by XOR-GMM. Red dots represent data with label $+1$, and blue dots represent data with label -1.

2.2. Two-Layer GCN

To discuss the learning dynamics of GCN, we define a GCN model as follows:

$$f(\boldsymbol{X}, \boldsymbol{A}) : \ \mathbb{R}^{N \times d} \ni \boldsymbol{X} \longmapsto \tilde{\boldsymbol{A}} \sigma(\boldsymbol{X}\boldsymbol{W}) \boldsymbol{a} \in \mathbb{R}^N$$

$$\tilde{\boldsymbol{A}} = \boldsymbol{D}^{-1}(\boldsymbol{A} + \boldsymbol{I}_N), \ \boldsymbol{D} = \boldsymbol{I}_N + \mathrm{diag}\left(\sum_i A_{1i}, \cdots, \sum_i A_{Ni} \right)$$

where σ is the activation function, we use the ReLU function in this paper. K is the number of neurons, $\boldsymbol{W} \in \mathbb{R}^{d \times K}$ is the first-layer weight matrix and $\boldsymbol{a} \in \mathbb{R}^K$ is the second-layer weight vector. We denote the jth column of $\boldsymbol{W}$ as $\boldsymbol{w}_j$, so $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K]$ and $\boldsymbol{w}_j \in \mathbb{R}^d$ for all $j \in [K]$. We assume the second-layer weight vector $\boldsymbol{a}$ is generated by $\mathrm{Unif}(\{\pm \frac{1}{\sqrt{K}}\}^d)$ and fixed during the training process. For simplicity, we sometimes omit the indices of $\boldsymbol{x}_i, y_i, \boldsymbol{w}_j$ and a_j. In addition, the sign of the i th component of $f(\boldsymbol{X}, \boldsymbol{A})$ corresponds to the prediction of the i th label. Note that the activation function is performed before the graph convolution operation to prevent loss of information during convolution because the features that have labels $+1$ and -1 are of the same mean $\mathbf{0}$ (this is mentioned in [4]).

2.3. Problem Settings

Our purpose is to predict labels from $\boldsymbol{X}, \boldsymbol{A}$ with high accuracy. To achieve this, we update the first-layer weight matrix $\boldsymbol{W}$ with online gradient descent. In each update step t, we generate N features and labels as $(\boldsymbol{y}^{(t)}, \boldsymbol{X}^{(t)}, \boldsymbol{A}^{(t)})$ for the training data from the same distribution as the test data. We define the empirical loss using the correlation loss

$$L(\boldsymbol{W}; \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{A}) = -\frac{1}{N}\sum_{i=1}^{N} y_i f(\boldsymbol{X}, \boldsymbol{A})_i$$

and update $\boldsymbol{W}$ by online gradient descent. To analyze the dynamics, we use the population expectation of gradients

$$\boldsymbol{W}^{(t+1)} = \boldsymbol{W}^{(t)} - \eta\, \mathbb{E}_{\boldsymbol{y}^{(t)}, \boldsymbol{X}^{(t)}, \boldsymbol{A}^{(t)}}[\nabla_{\boldsymbol{W}} L(\boldsymbol{W}^{(t)}; \boldsymbol{y}^{(t)}, \boldsymbol{X}^{(t)}, \boldsymbol{A}^{(t)})]$$

where η is the learning rate, t is the number of steps. In addition, $\boldsymbol{W}$ is initialized as $\boldsymbol{w}_j \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$ for all $j \in [K]$.

3. Learning Dynamics of GCN

In this section, we reveal the behavior of the learning dynamics in the GCN. To this end, we calculate the population expectation of the gradient and derive the neuron update equations.

3.1. Population Expectation of Online Gradient Descent

To discuss learning dynamics, we define three components of $\boldsymbol{w}$.

Definition 4 *For GCN defined in section 2, let $\boldsymbol{\mu}_{sig}$ and $\boldsymbol{\mu}_{opp}$ be defined as follows.*

$$\boldsymbol{\mu}_{sig} = \begin{cases} \boldsymbol{\mu}_1 & a\ \mathrm{sgn}(\alpha-\beta) > 0,\ \boldsymbol{\mu}_1^T\boldsymbol{w} > 0 \\ -\boldsymbol{\mu}_1 & a\ \mathrm{sgn}(\alpha-\beta) > 0,\ \boldsymbol{\mu}_1^T\boldsymbol{w} < 0 \\ \boldsymbol{\mu}_{-1} & a\ \mathrm{sgn}(\alpha-\beta) < 0,\ \boldsymbol{\mu}_{-1}^T\boldsymbol{w} > 0 \\ -\boldsymbol{\mu}_{-1} & a\ \mathrm{sgn}(\alpha-\beta) < 0,\ \boldsymbol{\mu}_{-1}^T\boldsymbol{w} < 0 \end{cases}, \quad \boldsymbol{\mu}_{opp} = \begin{cases} \boldsymbol{\mu}_{-1} & a\ \mathrm{sgn}(\alpha-\beta) > 0 \\ \boldsymbol{\mu}_1 & a\ \mathrm{sgn}(\alpha-\beta) < 0 \end{cases}.$$

In addition, define $\boldsymbol{w}_{sig}, \boldsymbol{w}_{opp}, \boldsymbol{w}_\perp$ using $\boldsymbol{\mu}_{sig}$ and $\boldsymbol{\mu}_{opp}$ as follows:

$$w_{sig} := \boldsymbol{\mu}_{sig}^T\boldsymbol{w},\ w_{opp} := \boldsymbol{\mu}_{opp}^T\boldsymbol{w}$$

$$\boldsymbol{w}_{sig} := w_{sig}\boldsymbol{\mu}_{sig},\ \boldsymbol{w}_{opp} := w_{opp}\boldsymbol{\mu}_{opp},\ \boldsymbol{w}_\perp := \boldsymbol{w} - \boldsymbol{w}_{sig} - \boldsymbol{w}_{opp}\ .$$

This decomposition is introduced to denote the signal learning on the XOR model. We want $\boldsymbol{w}_{sig}$ to grow and $\boldsymbol{w}_{opp}, \boldsymbol{w}_\perp$ to shrink, but we have to separate $\boldsymbol{w}_{opp}$ from $\boldsymbol{w}_\perp$ due to the symmetry of the XOR model. [6] utilized this decomposition to analyze the learning dynamics on the neural network with the XOR model and we apply it to GCN with XOR-CSBM.

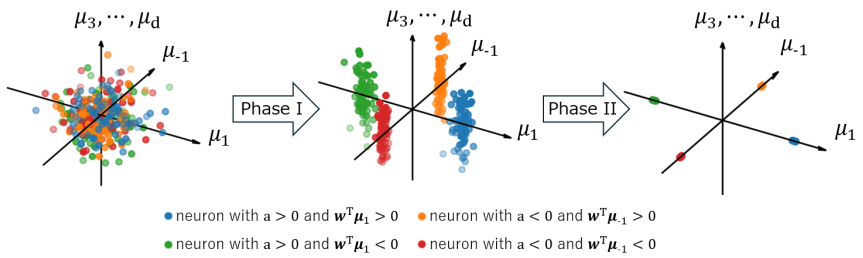


Figure 2: Illustration of the two-phase dynamics of GCN training. The vectors $\boldsymbol{\mu}_3, \cdots, \boldsymbol{\mu}_d$ form a basis for the orthogonal complement of the space spanned by $\boldsymbol{\mu}_{\pm 1}$. In Phase I only opposite-signal component shrinks and in Phase II the orthogonal component also decays.

Theorem 5 *Consider training the two-layer GCN as in section 2. In each learning step t, define $w_{sig}^{(t)}, w_{opp}^{(t)}, w_{\perp}^{(t)}$ as in definition 4. The population expectations of the gradient are*

$$\begin{cases} w_{sig}^{(t+1)} = w_{sig}^{(t)} + \eta g \left(\frac{\theta}{4\sqrt{K}} \mathrm{erf}\left(\frac{\theta w_{sig}^{(t)}}{\sqrt{2}\|\boldsymbol{w}^{(t)}\|} \right) - \gamma^{(t)} w_{sig}^{(t)} \right) \\ w_{opp}^{(t+1)} = w_{opp}^{(t)} + \eta g \left(-\frac{\theta}{4\sqrt{K}} \mathrm{erf}\left(\frac{\theta w_{opp}^{(t)}}{\sqrt{2}\|\boldsymbol{w}^{(t)}\|} \right) - \gamma^{(t)} w_{opp}^{(t)} \right) \\ \boldsymbol{w}_{\perp}^{(t+1)} = \boldsymbol{w}_{\perp}^{(t)} - \eta g \gamma^{(t)} \boldsymbol{w}_{\perp}^{(t)} \end{cases} \tag{1}$$

where $\gamma^{(t)}$ and g are

$$\gamma^{(t)} = \frac{-1}{\sqrt{8\pi K}\|\boldsymbol{w}^{(t)}\|} \left(\exp\left(\frac{-\theta^2 (w_{sig}^{(t)})^2}{2\|\boldsymbol{w}^{(t)}\|^2} \right) - \exp\left(\frac{-\theta^2 (w_{opp}^{(t)})^2}{2\|\boldsymbol{w}^{(t)}\|^2} \right) \right)$$

$$g = \mathrm{sgn}(\alpha - \beta) \left(\frac{\alpha - \beta}{\alpha + \beta} + \frac{4\beta}{N(\alpha+\beta)^2} \left(1 - \left(1 - \frac{\alpha+\beta}{2} \right)^N \right) \right)$$

Here, we define parameters $g, \gamma^{(t)}$. g is a constant that depends on α, β, which captures the effects of the graph structure, and $\gamma^{(t)}$ is an effective weight decay for each learning step.

Theorem 5 reveals a two-phase learning dynamics. In the first phase, the signal direction is separated from its opposite direction, while the orthogonal component remains nearly unchanged. In the second phase, the remaining non-signal directions are suppressed, and the neuron aligns with the signal direction. This separation is clearest in the small-learning-rate regime. Similar phases were observed for MLPs ([6]); we show that they also appear in GCNs. Figure 2 illustrates this behavior.

We state in detail analysis of the learning dynamics in section A.2. In addition, we apply this result to perfect recovery analysis and show that GCN outperforms MLP in section B.

4. Conclusion

In this paper, we analyzed the learning dynamics of online gradient descent for GCNs on XOR-CSBM. We showed that the training dynamics exhibit two distinct phases, through which the GCN eventually learns task-relevant nonlinear features.

REFERENCES

[1] Emmanuel Abbe, Jianqing Fan, and Kaizheng Wang. An ℓ_p theory of PCA and spectral clustering. *The Annals of Statistics*, 50(4):2359 – 2385, 2022.

[2] Pranjal Awasthi, Abhimanyu Das, and Sreenivas Gollapudi. A convergence analysis of gradient descent on graph neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 20385–20397. Curran Associates, Inc., 2021.

[3] Aseem Baranwal, Kimon Fountoulakis, and Aukosh Jagannath. Graph convolution for semi-supervised classification: Improved linear separability and out-of-distribution generalization, 2022.

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

[5] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems*, 28, 2015.

[6] Margalit Glasgow. SGD finds then tunes features in two-layer neural networks with near-optimal sample complexity: A case study in the XOR problem. In *The Twelfth International Conference on Learning Representations*, 2024.

[7] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983. ISSN 0378-8733.

[8] Wei Huang, Yuan Cao, Haonan Wang, Xin Cao, and Taiji Suzuki. Quantifying the optimization and generalization advantages of graph neural networks over multilayer perceptrons. In *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pages 2854–2862. PMLR, 03–05 May 2025.

[9] Haotian Ju, Dongyue Li, Aneesh Sharma, and Hongyang R. Zhang. Generalization in graph neural networks: Improved pac-bayesian bounds on graph diffusion. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 6314–6341. PMLR, 25–27 Apr 2023.

[10] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[11] Sanjukta Krishnagopal and Luana Ruiz. Graph neural tangent kernel: Convergence on large graphs. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17827–17841. PMLR, 23–29 Jul 2023.

[12] David Saad and Sara Solla. Dynamics of on-line gradient descent learning for multilayer neural networks. In *Advances in Neural Information Processing Systems*, volume 8. MIT Press, 1995.

[13] Haixiao Wang and Zhichao Wang. Optimal exact recovery in semi-supervised learning: A study of spectral methods and graph convolutional networks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 51614–51649. PMLR, 21–27 Jul 2024.

[14] Xinyi Wu, Zhengdao Chen, William Wei Wang, and Ali Jadbabaie. A non-asymptotic analysis of oversmoothing in graph neural networks. In *The Eleventh International Conference on Learning Representations*, 2023.

[15] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 974–983, 2018.

[16] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017.

Appendix A. Supplement of Learning Dynamics of GCN

In this section, we denote a key point of derivation of Theorem 5 and in detail derivation of learning dynamics phases stated in section 3.

A.1. Key Insight of derivation of Theorem 5

Here we mention an interesting decomposition that appears in the derivation of Theorem 5. To derive Theorem 5, we derive and utilize results that separate the influence of the graph when calculating gradients. In particular, we transform $\nabla_{\boldsymbol{w}} L$ as follows.

$$\nabla_{\boldsymbol{w}} L = -\frac{1}{N}\sum_{i=1}^{N} \underbrace{\left(\sum_{j=1}^{N} \frac{A_{ij} y_i y_j}{\sum_{k=1}^{N} A_{jk}}\right)}_{=:F_i(\boldsymbol{y},\boldsymbol{A})} y_i \sigma'(\boldsymbol{x}_i^T \boldsymbol{w}) a \boldsymbol{x}_i$$

$$\mathbb{E}_{\boldsymbol{X},\boldsymbol{y},\boldsymbol{A}}[\nabla_{\boldsymbol{w}} L] = -\mathbb{E}_{\boldsymbol{y}}\left[\frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{\boldsymbol{A}}[F_i(\boldsymbol{y},\boldsymbol{A})|\,\boldsymbol{y}]\, \mathbb{E}_{\boldsymbol{x}_i}[y_i \sigma'(\boldsymbol{x}_i^T \boldsymbol{w}) a \boldsymbol{x}_i|\,\boldsymbol{y}]\right]$$

Here we used the fact that $\boldsymbol{A}$ and $\boldsymbol{X}$ are independent when given $\boldsymbol{y}$ from the definition of XOR-CSBM. From this decomposition, the graph structure influences the learning dynamics through the expectation of $F_i(\boldsymbol{y}, \boldsymbol{A})$.

From this decomposition, one important extension is to study more general data-generating models in which $\boldsymbol{X}$ and $\boldsymbol{A}$ are dependent, since covariance terms between them may lead to qualitatively different learning dynamics. This paper considers XOR-CSBM, in which $\boldsymbol{X}$ and $\boldsymbol{A}$ are independent; extending this to systems where $\boldsymbol{X}$ and $\boldsymbol{A}$ are not independent is left as future work.

A.2. Derivation of Learning Dynamics Phases from Theorem 5

In this section, we denote how the phases of learning dynamics mentioned in section 3 are derived from Theorem 5.

Phase I: separation of the signal and opposite-signal directions. We initialize each neuron as $\boldsymbol{w}_j \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$, so $w_{sig}^{(t)}/\|\boldsymbol{w}^{(t)}\| \simeq w_{opp}^{(t)}/\|\boldsymbol{w}^{(t)}\| \simeq 1/\sqrt{d}$ when t is small. In this regime $\gamma^{(t)} \simeq 0$ and thus the update rule in (1) is simplified as

$$\begin{cases} w_{sig}^{(t+1)} = w_{sig}^{(t)} + \eta g \frac{\theta^2}{\sqrt{8\pi K}\|\boldsymbol{w}^{(t)}\|} w_{sig}^{(t)} \\ w_{opp}^{(t+1)} = w_{opp}^{(t)} - \eta g \frac{\theta^2}{\sqrt{8\pi K}\|\boldsymbol{w}^{(t)}\|} w_{opp}^{(t)} \\ \boldsymbol{w}_\perp^{(t+1)} = \boldsymbol{w}_\perp^{(t)} \end{cases}.$$

In this phase, $\boldsymbol{w}_\perp$ barely changes, so $\|\boldsymbol{w}^{(t)}\|$ remains close to its initial value. The above equations therefore show a clear asymmetry: the signal component w_{sig} is amplified, whereas the opposite-signal component w_{opp} is suppressed. In particular, under the small-learning-rate condition $\eta \lesssim \sqrt{K}/g\theta^2$, the update for w_{opp} is contractive, and hence w_{opp} decays exponentially while w_{sig} grows exponentially. Thus, the first phase acts as a coarse selection stage, in which the dynamics distinguishes the correct signal direction from its opposite direction, while leaving the orthogonal component essentially untouched.

This clean separation of phases relies on the learning rate being sufficiently small. When the learning rate is larger, this clean separation of timescales may no longer hold, and the two phases need not be sharply distinguishable.

This regime continues until the signal component becomes non-negligible compared with the norm of the neuron. Indeed, from the Phase I dynamics,

$$\frac{w_{sig}^{(t)}}{\|\boldsymbol{w}^{(t)}\|} \simeq \frac{1}{\sqrt{d}} \left(1 + \frac{\eta g \theta^2}{\sqrt{8\pi K}}\right)^t,$$

so after $O\left(\frac{\sqrt{K}\log d}{\eta g \theta^2}\right)$ steps, the ratio $w_{sig}^{(t)}/\|\boldsymbol{w}^{(t)}\|$ reaches constant order. At this point, the training leaves the initial linearized regime and enters a second phase in which the nonlinear effect becomes substantial.

Phase II: elimination of non-signal directions and alignment. Once $w_{sig}^{(t)}$ becomes dominant over $w_{opp}^{(t)}$, we have $w_{opp}^{(t)}/\|\boldsymbol{w}^{(t)}\| \ll 1$ and $w_{sig}^{(t)} \gg w_{opp}^{(t)}$, so

$$\gamma^{(t)} \simeq \frac{1}{\sqrt{8\pi K}\|\boldsymbol{w}^{(t)}\|} \left(1 - \exp\left(\frac{-\theta^2 (w_{sig}^{(t)})^2}{2\|\boldsymbol{w}^{(t)}\|^2}\right)\right).$$

Then the update rule (1) becomes

$$\begin{cases} w_{sig}^{(t+1)} = w_{sig}^{(t)} + \eta g \frac{1}{\sqrt{8\pi K}} \left(\frac{2\theta}{\sqrt{\pi}} \mathrm{erf}\left(\frac{\theta w_{sig}^{(t)}}{\sqrt{2}\|\boldsymbol{w}^{(t)}\|}\right) - \frac{w_{sig}^{(t)}}{\|\boldsymbol{w}^{(t)}\|}\left(1 - \exp\left(\frac{-\theta^2 (w_{sig}^{(t)})^2}{2\|\boldsymbol{w}^{(t)}\|^2}\right)\right)\right) \\ w_{opp}^{(t+1)} = w_{opp}^{(t)} - \eta g \frac{1}{\sqrt{8\pi K}} \left(\theta^2 + 1 - \exp\left(\frac{-\theta^2 (w_{sig}^{(t)})^2}{2\|\boldsymbol{w}^{(t)}\|^2}\right)\right) \frac{w_{opp}^{(t)}}{\|\boldsymbol{w}^{(t)}\|} \\ \boldsymbol{w}_\perp^{(t+1)} = \boldsymbol{w}_\perp^{(t)} - \eta g \frac{1}{\sqrt{8\pi K}} \left(1 - \exp\left(\frac{-\theta^2 (w_{sig}^{(t)})^2}{2\|\boldsymbol{w}^{(t)}\|^2}\right)\right) \frac{\boldsymbol{w}_\perp^{(t)}}{\|\boldsymbol{w}^{(t)}\|} \end{cases}.$$

The qualitative behavior now changes in an important way. While w_{sig} continues to increase, both w_{opp} and the orthogonal component $\boldsymbol{w}_\perp$ are driven toward zero. In contrast to Phase I, where only the opposite-signal direction is suppressed, the second phase also removes directions irrelevant to the task. Moreover, when $0 < w_{sig}/\|\boldsymbol{w}\| \leq 1$ and $\theta > 0$, the increment $w_{sig}^{(t+1)} - w_{sig}^{(t)}$ never vanishes. Hence w_{sig} continues to grow and eventually diverges to $+\infty$.

Taken together, the two phases reveal a simple mechanism of feature learning: the dynamics first amplify the signal component against its opposite direction, and then eliminate the remaining orthogonal noise. This directly implies the following corollary.

Corollary 6 *After sufficiently many steps of online gradient descent, the signal component w_{sig} dominates both w_{opp} and $\|\boldsymbol{w}_\perp\|$ for every neuron. In other words, each neuron weight $\boldsymbol{w}$ aligns with the signal direction.*

Appendix B. Application to Perfect Recovery Analysis: Advantage of GCN

In this section, we consider perfect recovery from data generated by XOR-CSBM. First, we demonstrate the existence of a parameter space where perfect recovery cannot be achieved by any algorithm. Next, we show that in all other parameter spaces, GCN achieves perfect recovery, whereas MLP (defined eliminated graph convolution layer from GCN) does not.

B.1. Additional Setup

Before discussion, we define the perfect recovery of the estimator, Two-Layer MLP and Two-Layer GCN with weighted self loop.

Definition 7 (Perfect Recovery) *For $(\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{A})$ an estimator $\hat{\boldsymbol{y}}(\boldsymbol{X}, \boldsymbol{A})$ achieves perfect recovery when*

$$\lim_{N\to\infty} \mathbb{P}(\hat{\boldsymbol{y}}(\boldsymbol{X}, \boldsymbol{A}) = \pm\boldsymbol{y}) = 1 \quad .$$

Definition 8 (Two-Layer MLP and Two-Layer GCN with weighted self loop)

1. *We define a Two-Layer MLP as follows.*

$$f^{\mathrm{MLP}}(\boldsymbol{X}) : \mathbb{R}^{N\times d} \ni \boldsymbol{X} \longmapsto \sigma(\boldsymbol{X}\boldsymbol{W})\boldsymbol{a} \in \mathbb{R}^N$$

2. *We define a Two-Layer GCN with weighted self loop as follows.*

$$f^{\mathrm{GCN}}(\boldsymbol{X}, \boldsymbol{A}) : \mathbb{R}^{N\times d} \ni \boldsymbol{X} \longmapsto \tilde{\boldsymbol{A}}\sigma(\mathbf{X}\mathbf{W})\boldsymbol{a} \in \mathbb{R}^N$$

where

$$\tilde{\boldsymbol{A}} = \boldsymbol{D}^{-1}(\boldsymbol{A} + \rho\boldsymbol{I}_N),\ \boldsymbol{D} = \rho\boldsymbol{I}_N + diag\left(\sum_i A_{1i}, \cdots, \sum_i A_{Ni}\right) \quad .$$

Here ρ is the weight of the self loops.

This definition of the MLP is derived from the GCN definition by removing the graph convolutional layer, and we will use it to compare the MLP and the GCN. This comparison is similar to [8]. In addition, we define the loss for MLP as follows.

$$L^{\mathrm{MLP}}(\boldsymbol{W}; \boldsymbol{y}, \boldsymbol{X}) = -\frac{1}{N}\sum_{i=1}^{N} y_i f^{\mathrm{MLP}}(\boldsymbol{X})_i$$

For simplicity, we sometimes denote f by both f^{MLP} and f^{GCN}, and L by both L^{MLP} and L^{GCN}.

B.2. Impossibility of Perfect Recovery

To discuss perfect recovery of XOR-CSBM, we assume the scaling of the parameters.

Assumption 1 *Consider $N, K \to \infty$, $q_N = \log N$, α, β and θ are scaled as $\alpha = a \cdot \frac{q_N}{N}, \beta = b \cdot \frac{q_N}{N}, \theta^2/q_N = c$ with $a, b, c = O(1)$.*

This scaling is similar to [13], [1], [14]. If we assume this assumption, the sufficiency of the impossibility to perfect recovery will be derived.

Theorem 9 *Under assumption 1, any estimator will miss-classify when*

$$I(a, b, c) := \frac{(\sqrt{a} - \sqrt{b})^2 + c}{2} < 1\ .$$

To prove Theorem 9, we consider a Maximum Likelihood Estimator (MLE). MLE is the best estimator for a classification task so if MLE fails perfect recovery any estimator fails, and when $I(a, b, c) < 1$ there is a $\boldsymbol{y}'$ with high probability such that $\boldsymbol{y}' \neq \boldsymbol{y}$ and the likelihood of $\boldsymbol{y}'$ is larger than that of $\boldsymbol{y}$. This idea is similar to [13].

B.3. Advantage of GCN against MLP

We consider training the inner weights of MLP and GCN as online gradient descent as in section 2, and additionally, we consider adding weighted self loop to the adjacency matrix after inner weight learning. We denote the algorithms in Figure 1 and Figure 2. This weight balances the relative importance between the features of a node and those of its neighbors when calculating its output. This is mentioned in [10] and [13] optimized it in GCN with CSBM.

To analyze the condition of perfect recovery, we simplify the distribution of $(\boldsymbol{w}, a)$ and the degree matrix in the GCN.

Assumption 2 *After the online gradient descent learning of $\boldsymbol{W}$, each $(\boldsymbol{w}, a)$ of the MLP becomes one of $(w_{sig}\boldsymbol{\mu}_1, \frac{1}{\sqrt{K}})$, $(-w_{sig}\boldsymbol{\mu}_1, \frac{1}{\sqrt{K}})$, $(w_{sig}\boldsymbol{\mu}_{-1}, \frac{-1}{\sqrt{K}})$ and $(-w_{sig}\boldsymbol{\mu}_{-1}, \frac{-1}{\sqrt{K}})$, each $(\boldsymbol{w}, a)$ of the GCN becomes one of $(w_{sig}\boldsymbol{\mu}_1, \frac{\mathrm{sgn}(\alpha-\beta)}{\sqrt{K}})$, $(-w_{sig}\boldsymbol{\mu}_1, \frac{\mathrm{sgn}(\alpha-\beta)}{\sqrt{K}})$, $(w_{sig}\boldsymbol{\mu}_{-1}, \frac{-\mathrm{sgn}(\alpha-\beta)}{\sqrt{K}})$ and $(-w_{sig}\boldsymbol{\mu}_{-1}, \frac{-\mathrm{sgn}(\alpha-\beta)}{\sqrt{K}})$. The numbers for each patterns of $(\boldsymbol{w}, a)$ are all the same $(= \frac{K}{4})$. Also, we consider the adjusted matrix with weighted self loops $\boldsymbol{A} + \rho\boldsymbol{I}$ and simplify the degree matrix with weighted self loops $\boldsymbol{D}$ in the GCN model as $\boldsymbol{D} = D\boldsymbol{I}_N, D = \frac{(\alpha+\beta)N}{2} + \rho = \frac{(a+b)q_N}{2} + \rho$.*

From Corollary 6 and the initialization of $\boldsymbol{W}, \boldsymbol{a}$, the pair of $(\boldsymbol{w}, a)$ will converge to each pattern with the same probability. Now K is sufficiently large from Assumption 1 the difference of the number of each patterns will be negligible. Also about this assumption of the degree matrix, we now consider that the average degree has $O(q_N) = O(\log N)$ so it will be satisfied when $N \to \infty$.

Theorem 10 *We consider MLP and GCN defined in section 2. Under assumptions 1 and 2,*

(i) The MLP after learning with algorithm 1 achieves perfect recovery with $c/2 > 1$.
(ii) The GCN after learning with algorithm 2 with self loop $\rho = \frac{2c}{\log(a/b)} q_N$ achieves perfect recovery when

$$\frac{(\sqrt{a} - \sqrt{b})^2 + c}{2} = I(a, b, c) > 1 \, .$$

From Theorem 10, when $\frac{(\sqrt{a}-\sqrt{b})^2+c}{2} > 1$ and $\frac{c}{2} < 1$ GCN can achieve perfect recovery while MLP fails, so GCN outperforms MLP by leveraging the graph structure. And from Theorem 9 and 10 GCN with optimal weighted self loop becomes one of the best estimators, which is consistent with Theorem 3.9 of [13].

Appendix C. Expectations of the loss gradients

C.1. Proof of Theorem 5

$$\begin{aligned}
\nabla_{\boldsymbol{w}_j} L &= -\frac{1}{N} \sum_{i=1}^{N} \frac{y_i}{|\mathcal{N}(i)|} \sum_{k \in \mathcal{N}(i)} \sigma'(\boldsymbol{x}_k^T \boldsymbol{w}_j) a_j \boldsymbol{x}_k \\
&= -\frac{1}{N} \sum_{k=1}^{N} \sum_{i=1}^{N} \frac{A_{ik} y_k}{\sum_{j=1}^{N} A_{jk}} \sigma'(\boldsymbol{x}_i^T \boldsymbol{w}_j) a_j \boldsymbol{x}_i \\
&= -\frac{1}{N} \sum_{i=1}^{N} \left(\sum_{k=1}^{N} \frac{A_{ik} y_i y_k}{\sum_{j=1}^{N} A_{jk}} \right) y_i \sigma'(\boldsymbol{x}_i^T \boldsymbol{w}_j) a_j \boldsymbol{x}_i = -\frac{1}{N} \sum_{i=1}^{N} F_i(\boldsymbol{y}, \boldsymbol{A}) y_i \sigma'(\boldsymbol{x}_i^T \boldsymbol{w}_j) a_j \boldsymbol{x}_i
\end{aligned}$$

Here we denote $\sum_{j=1}^{N} \frac{A_{ij} y_i y_j}{\sum_{k=1}^{N} A_{jk}}$ by $F_i(\boldsymbol{y}, \boldsymbol{A})$.

$$\begin{aligned}
-\mathbb{E}_{\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{A}}[\nabla_{\boldsymbol{w}} L] &= \mathbb{E}_{\boldsymbol{x}_1, \boldsymbol{y}, \boldsymbol{A}}[F_i(\boldsymbol{y}, \boldsymbol{A}) y_i \sigma'(\boldsymbol{x}_1^T \boldsymbol{w}) a \boldsymbol{x}_1] \\
&= \frac{1}{2} \mathbb{E}_{\boldsymbol{x}_1, y_2, \cdots, y_N, \boldsymbol{A}}[F_1(\boldsymbol{y}, \boldsymbol{A}) y_1 \sigma'(\boldsymbol{x}_1^T \boldsymbol{w}) a \boldsymbol{x}_1 | y_1 = 1] \\
&\quad + \frac{1}{2} \mathbb{E}_{\boldsymbol{x}_1, y_2, \cdots, y_N, \boldsymbol{A}}[F_1(\boldsymbol{y}, \boldsymbol{A}) y_1 \sigma'(\boldsymbol{x}_1^T \boldsymbol{w}) a \boldsymbol{x}_1 | y_1 = -1] \\
&= \frac{1}{2} \mathbb{E}_{y_2, \cdots, y_N, \boldsymbol{A}}[F_1(\boldsymbol{y}, \boldsymbol{A}) | y_1 = 1] \mathbb{E}_{\boldsymbol{x}_1}[y_1 \sigma'(\boldsymbol{x}_1^T \boldsymbol{w}) a \boldsymbol{x}_1 | y_1 = 1] \\
&\quad + \frac{1}{2} \mathbb{E}_{y_2, \cdots, y_N, \boldsymbol{A}}[F_1(\boldsymbol{y}, \boldsymbol{A}) | y_1 = -1] \mathbb{E}_{\boldsymbol{x}_1}[y_1 \sigma'(\boldsymbol{x}_1^T \boldsymbol{w}) a \boldsymbol{x}_1 | y_1 = -1]
\end{aligned}$$

The following identity holds for the expectation of the expectation of $F_1(\boldsymbol{y}, \boldsymbol{A})$.

Lemma 11 *When* $\boldsymbol{y} \sim \mathrm{Unif}(\{\pm 1\}^N)$, $\boldsymbol{A} \sim \mathrm{SBM}(\boldsymbol{y}, \alpha, \beta)$,

$$\begin{aligned}\mathbb{E}_{y_2,\cdots,y_N,\boldsymbol{A}}[F_1(\boldsymbol{y},\boldsymbol{A})|y_1=1] &= \mathbb{E}_{y_2,\cdots,y_N,\boldsymbol{A}}[F_1(\boldsymbol{y},\boldsymbol{A})|y_1=-1] \\ &= \frac{\alpha-\beta}{\alpha+\beta}+\frac{4\beta}{N(\alpha+\beta)^2}\left(1-\left(1-\frac{\alpha+\beta}{2}\right)^N\right)\end{aligned}$$

So, $\mathbb{E}_{y_2,\cdots,y_N,\boldsymbol{A}}[F_1(\boldsymbol{y},\boldsymbol{A})|y_1=1] = \mathbb{E}_{y_2,\cdots,y_N,\boldsymbol{A}}[F_1(\boldsymbol{y},\boldsymbol{A})|y_1=-1] = g\ \mathrm{sgn}(\alpha-\beta)$ and

$$\begin{aligned}-\mathbb{E}_{\boldsymbol{X},\boldsymbol{y},\boldsymbol{A}}[\nabla_{\boldsymbol{w}} L] &= g\ \mathrm{sgn}(\alpha-\beta)\frac{1}{2}(\mathbb{E}_{\boldsymbol{x}}[y\sigma'(\boldsymbol{x}^T\boldsymbol{w})a\boldsymbol{x}|y=1]+\mathbb{E}_{\boldsymbol{x}_1}[y\sigma'(\boldsymbol{x}^T\boldsymbol{w})a\boldsymbol{x}|y=-1]) \\ &= g\ a\ \mathrm{sgn}(\alpha-\beta)\mathbb{E}_{\boldsymbol{x},y}[y\sigma'(\boldsymbol{x}^T\boldsymbol{w})\boldsymbol{x}]\end{aligned}$$

In addition, the following identity holds for the part of the expectation by $\boldsymbol{x}$.

Lemma 12 *Suppose that* $y \sim \mathrm{Unif}(\{\pm 1\}), \boldsymbol{x} \sim \mathrm{XOR}(\boldsymbol{\mu}_1, \boldsymbol{\mu}_{-1}, y, \theta)$. *Let* $w_{\pm 1} = \boldsymbol{\mu}_{\pm 1}^T \boldsymbol{w}$ *be the components of* $\boldsymbol{\mu}_{\pm 1}$ *of* $\boldsymbol{w}$, $\boldsymbol{e}_w = \boldsymbol{w}/\|\boldsymbol{w}\|$ *be the unit vector along* $\boldsymbol{w}$, *and*

$$\begin{aligned}\mathbb{E}_{\boldsymbol{x},y}[y\sigma'(\boldsymbol{x}^T\boldsymbol{w})\boldsymbol{x}] &= \frac{1}{4}\left(\theta\ \mathrm{erf}\left(\frac{\theta w_1}{\sqrt{2}\|\boldsymbol{w}\|}\right)\boldsymbol{\mu}_1-\theta\ \mathrm{erf}\left(\frac{\theta w_{-1}}{\sqrt{2}\|\boldsymbol{w}\|}\right)\boldsymbol{\mu}_{-1}\right. \\ &\quad \left.+\sqrt{\frac{2}{\pi}}\left(\exp\left(\frac{-\theta^2 w_1^2}{2\|\boldsymbol{w}\|^2}\right)-\exp\left(\frac{-\theta^2 w_{-1}^2}{2\|\boldsymbol{w}\|^2}\right)\right)\boldsymbol{e}_w\right)\end{aligned}$$

Therefore, the expectation of the gradient is

$$\begin{aligned}-\mathbb{E}_{\boldsymbol{X},\boldsymbol{y},\boldsymbol{A}}[\nabla_{\boldsymbol{w}} L] &= \frac{1}{4}g\ a\ \mathrm{sgn}(\alpha-\beta)\left(\theta\ \mathrm{erf}\left(\frac{\theta w_1}{\sqrt{2}\|\boldsymbol{w}\|}\right)\boldsymbol{\mu}_1-\theta\ \mathrm{erf}\left(\frac{\theta w_{-1}}{\sqrt{2}\|\boldsymbol{w}\|}\right)\boldsymbol{\mu}_{-1}\right. \\ &\quad \left.+\sqrt{\frac{2}{\pi}}\left(\exp\left(\frac{-\theta^2 w_1^2}{2\|\boldsymbol{w}\|^2}\right)-\exp\left(\frac{-\theta^2 w_{-1}^2}{2\|\boldsymbol{w}\|^2}\right)\right)\boldsymbol{e}_w\right) \\ &= g\ a\sqrt{K}\ \mathrm{sgn}(\alpha-\beta)\left(\left(\frac{\theta}{4\sqrt{K}}\mathrm{erf}\left(\frac{\theta w_1}{\sqrt{2}\|\boldsymbol{w}\|}\right)-\gamma w_1\right)\boldsymbol{\mu}_1\right. \\ &\quad \left.+\left(-\frac{\theta}{4\sqrt{K}}\mathrm{erf}\left(\frac{\theta w_{-1}}{\sqrt{2}\|\boldsymbol{w}\|}\right)-\gamma w_{-1}\right)\boldsymbol{\mu}_{-1}-\gamma\boldsymbol{w}_\perp\right)\end{aligned}$$

Here we denote

$$\gamma = -\frac{1}{\sqrt{8\pi K}\|\boldsymbol{w}\|}\left(\exp\left(\frac{-\theta^2 w_1^2}{2\|\boldsymbol{w}\|^2}\right)-\exp\left(\frac{-\theta^2 w_{-1}^2}{2\|\boldsymbol{w}\|^2}\right)\right)$$

From definition 4, this equation is equal to equation (1) for all the patterns of $\boldsymbol{\mu}_{sig}, \boldsymbol{\mu}_{opp}$. Also, when the MLP

$$\begin{aligned}
-\mathbb{E}_{\boldsymbol{X},\boldsymbol{y}}[\nabla_{\boldsymbol{w}} L^{\mathrm{MLP}}] &= \mathbb{E}_{\boldsymbol{X},\boldsymbol{y}}\left[\frac{1}{N}\sum_{i=1}^{N} y_i \sigma'(\boldsymbol{x}_i^T \boldsymbol{w}) a \boldsymbol{x}_i\right] \\
&= a\mathbb{E}_{\boldsymbol{x},y}\left[y\sigma'(\boldsymbol{x}^T\boldsymbol{w})\boldsymbol{x}\right] \\
&= \frac{a}{4}\left(\theta\,\mathrm{erf}\left(\frac{\theta w_1}{\sqrt{2}\|\boldsymbol{w}\|}\right)\boldsymbol{\mu}_1 - \theta\,\mathrm{erf}\left(\frac{\theta w_{-1}}{\sqrt{2}\|\boldsymbol{w}\|}\right)\boldsymbol{\mu}_{-1}\right. \\
&\quad \left. + \sqrt{\frac{2}{\pi}}\left(\exp\left(\frac{-\theta^2 w_1^2}{2\|\boldsymbol{w}\|^2}\right) - \exp\left(\frac{-\theta^2 w_{-1}^2}{2\|\boldsymbol{w}\|^2}\right)\right)\boldsymbol{e}_w\right)
\end{aligned}$$

this is equal to the result of the GCN when $g = 1, \mathrm{sgn}(\alpha - \beta) = 1$. In conclusion, the proof of theorem 5 is completed.

Proof [proof of lemma 11]

$$\begin{aligned}
\mathbb{E}_{y_2,\cdots,y_N,\boldsymbol{A}}[F_1(\boldsymbol{y},\boldsymbol{A})|y_1 = 1] &= \mathbb{E}_{y_2,\cdots,y_N,\boldsymbol{A}}\left[\frac{1}{\sum_{j=1}^{N} A_{j1}}\middle| y_1 = 1\right] \\
&\quad + (N-1)\mathbb{E}_{y_2,\cdots,y_N,\boldsymbol{A}}\left[\frac{A_{12}y_2}{\sum_{j=1}^{N} A_{j2}}\middle| y_1 = 1\right] \\
&= \mathbb{E}_{y_2,\cdots,y_N,\boldsymbol{A}}\left[\frac{1}{1+\sum_{j=2}^{N} A_{1j}}\middle| y_1 = 1\right] \\
&\quad + \frac{(N-1)\alpha}{2}\,\mathbb{E}_{y_3,\cdots,y_N,\boldsymbol{A}}\left[\frac{1}{2+\sum_{j=3}^{N} A_{2j}}\middle| y_2 = 1\right] \\
&\quad + \frac{(N-1)\beta}{2}\,\mathbb{E}_{y_3,\cdots,y_N,\boldsymbol{A}}\left[\frac{-1}{2+\sum_{j=3}^{N} A_{2j}}\middle| y_2 = -1\right] \\
&= \mathbb{E}_{y_2,\cdots,y_N,\boldsymbol{A}}\left[\frac{1}{1+\sum_{j=2}^{N} A_{1j}}\middle| y_1 = 1\right] \\
&\quad + \frac{(N-1)(\alpha-\beta)}{2}\,\mathbb{E}_{y_3,\cdots,y_N,\boldsymbol{A}}\left[\frac{1}{2+\sum_{j=3}^{N} A_{2j}}\middle| y_2 = 1\right]
\end{aligned}$$

Since when given y_i $A_{ij} \sim \mathrm{Ber}(\frac{\alpha+\beta}{2})$, we denote $p = \frac{\alpha+\beta}{2}$ and

$$\begin{aligned}
\mathbb{E}_{y_2,\cdots,y_N,\boldsymbol{A}}\left[\frac{1}{1+\sum_{j=2}^{N} A_{1j}}\middle| y_1 = 1\right] &= \sum_{n=0}^{N-1}\binom{N-1}{n} p^n (1-p)^{N-1-n}\frac{1}{1+n} \\
&= \sum_{n=0}^{N-1}\binom{N-1}{n} p^n (1-p)^{N-1-n}\int_0^1 x^n dx \\
&= \int_0^1 (px+1-p)^{N-1} dx = \frac{1}{Np}\left(1-(1-p)^N\right) \\
\mathbb{E}_{y_3,\cdots,y_N,\boldsymbol{A}}\left[\frac{1}{1+\sum_{j=3}^{N} A_{2j}}\middle| y_2 = 1\right] &= \sum_{n=0}^{N-2}\binom{N-2}{n} p^n (1-p)^{N-2-n}\frac{1}{2+n} \\
&= \sum_{n=0}^{N-2}\binom{N-2}{n} p^n (1-p)^{N-2-n}\int_0^1 x^{n+1} dx \\
&= \int_0^1 x\,(px+1-p)^{N-1} dx \\
&= \frac{1}{(N-1)p}\left(1-\frac{1}{Np}\left(1-(1-p)^N\right)\right)
\end{aligned}$$

These results do not depend on y_1, therefore

$$\begin{aligned}
\mathbb{E}_{y_2,\cdots,y_N,\boldsymbol{A}}[F_1(\boldsymbol{y},\boldsymbol{A})|y_1=1] &= \mathbb{E}_{y_2,\cdots,y_N,\boldsymbol{A}}[F_1(\boldsymbol{y},\boldsymbol{A})|y_1=-1] \\
&= \frac{\alpha-\beta}{\alpha+\beta} + \frac{4\beta}{N(\alpha+\beta)^2}\left(1-\left(1-\frac{\alpha+\beta}{2}\right)^N\right)
\end{aligned}$$

■

Proof [proof of lemma 12] We denote $z_w = \boldsymbol{e}_w^T \boldsymbol{z}$ and

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{x}}[y\sigma'(\boldsymbol{x}^T\boldsymbol{w})\boldsymbol{x}] &= \frac{1}{4}(\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)}[\sigma'((\theta\boldsymbol{\mu}_1+\boldsymbol{z})^T\boldsymbol{w})(\theta\boldsymbol{\mu}_1+\boldsymbol{z})] \\
&\quad +\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)}[\sigma'((-\theta\boldsymbol{\mu}_1+\boldsymbol{z})^T\boldsymbol{w})(-\theta\boldsymbol{\mu}_1+\boldsymbol{z})] \\
&\quad +\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)}[-\sigma'((\theta\boldsymbol{\mu}_{-1}+\boldsymbol{z})^T\boldsymbol{w})(\theta\boldsymbol{\mu}_{-1}+\boldsymbol{z})] \\
&\quad +\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)}[-\sigma'((-\theta\boldsymbol{\mu}_{-1}+\boldsymbol{z})^T\boldsymbol{w})(-\theta\boldsymbol{\mu}_{-1}+\boldsymbol{z})]) \\
&= \frac{1}{4}(\mathbb{E}_{z_w\sim\mathcal{N}(0,1)}[\sigma'(\theta w_1 + z_w\|\boldsymbol{w}\|) - \sigma'(-\theta w_1 + z_w\|\boldsymbol{w}\|)]\theta\boldsymbol{\mu}_1 \\
&\quad -\mathbb{E}_{z_w\sim\mathcal{N}(0,1)}[\sigma'(\theta w_{-1} + z_w\|\boldsymbol{w}\|) - \sigma'(-\theta w_{-1} + z_w\|\boldsymbol{w}\|)]\theta\boldsymbol{\mu}_{-1} \\
&\quad +\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)}[(\sigma'(\theta w_1 + z_w\|\boldsymbol{w}\|) + \sigma'(-\theta w_1 + z_w\|\boldsymbol{w}\|))\boldsymbol{z}] \\
&\quad -\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)}[(\sigma'(\theta w_{-1} + z_w\|\boldsymbol{w}\|) + \sigma'(-\theta w_{-1} + z_w\|\boldsymbol{w}\|))\boldsymbol{z}]) \\
&= \frac{1}{4}\left(\theta\,\mathrm{erf}\left(\frac{\theta w_1}{\sqrt{2}\|\boldsymbol{w}\|}\right)\boldsymbol{\mu}_1 - \theta\,\mathrm{erf}\left(\frac{\theta w_{-1}}{\sqrt{2}\|\boldsymbol{w}\|}\right)\boldsymbol{\mu}_{-1}\right. \\
&\quad \left. +\sqrt{\frac{2}{\pi}}\left(\exp\left(\frac{-\theta^2 w_1^2}{2\|\boldsymbol{w}\|^2}\right) - \exp\left(\frac{-\theta^2 w_{-1}^2}{2\|\boldsymbol{w}\|^2}\right)\right)\boldsymbol{e}_w\right)
\end{aligned}$$

Here we used these formulas for a random variable $z \sim \mathcal{N}(0,1)$ and any constants C, D

$$\mathbb{E}_{z\sim\mathcal{N}(0,1)}[\sigma'(C+Dz) - \sigma'(-C+Dz)] = \mathrm{erf}\left(\frac{C}{\sqrt{2}D}\right)$$

$$\mathbb{E}_{z\sim\mathcal{N}(0,1)}[(\sigma'(C+Dz) + \sigma'(-C+Dz))z] = \sqrt{\frac{2}{\pi}}\exp\left(-\frac{C^2}{2D^2}\right).$$

■

Appendix D. Information Threshold of Perfect Recovery for XOR-CSBM

We denote the true labels as $\boldsymbol{y}^*$ and the estimator as $\hat{\boldsymbol{y}}$. The probability of failing the perfect recovery is

$$\begin{aligned}\mathbb{P}_{\text{fail}} := \mathbb{P}(\hat{\boldsymbol{y}} \neq \pm\boldsymbol{y}^*) &= \sum_{\boldsymbol{A},\boldsymbol{X}} (1 - \mathbb{P}(\hat{\boldsymbol{y}} = \pm\boldsymbol{y}^* | \boldsymbol{A}, \boldsymbol{X})) \cdot \mathbb{P}(\boldsymbol{A}, \boldsymbol{X}) \\ &= 1 - \sum_{\boldsymbol{A},\boldsymbol{X}} \mathbb{P}(\boldsymbol{A}, \boldsymbol{X} | \hat{\boldsymbol{y}} = \pm\boldsymbol{y}^*) \cdot \mathbb{P}(\hat{\boldsymbol{y}} = \pm\boldsymbol{y}^*)\end{aligned}$$

Because $\boldsymbol{y}^*$ is generated with a uniform distribution in $\{\pm1\}^d$, the estimator that minimizes $\mathbb{P}_{\text{fail}}$ is the maximum likelihood estimator $\hat{\boldsymbol{y}}^{\text{MLP}}$:

$$\begin{aligned}\hat{\boldsymbol{y}}^{\text{MLP}} &= \operatorname*{argmax}_{\boldsymbol{z}\in\{\pm1\}^d} \mathbb{P}(\boldsymbol{A}, \boldsymbol{X} | \boldsymbol{y} = \boldsymbol{z}) \\ &= \operatorname*{argmax}_{\boldsymbol{z}\in\{\pm1\}^N} \mathbb{P}(\boldsymbol{A} | \boldsymbol{y} = \boldsymbol{z}) \cdot \mathbb{P}(\boldsymbol{X} | \boldsymbol{y} = \boldsymbol{z}) \\ &= \operatorname*{argmax}_{\boldsymbol{z}\in\{\pm1\}^N} \log\mathbb{P}(\boldsymbol{A} | \boldsymbol{y} = \boldsymbol{z}) + \log\mathbb{P}(\boldsymbol{X} | \boldsymbol{y} = \boldsymbol{z}) =: \operatorname*{argmax}_{\boldsymbol{z}\in\{\pm1\}^N} f(\boldsymbol{z})\end{aligned}$$

here we used the fact that $\boldsymbol{A}$ and $\boldsymbol{X}$ are independent when given the label because $(\boldsymbol{X}, \boldsymbol{A}) \sim$ XOR-CSBM. $\hat{\boldsymbol{y}}^{\text{MLP}}$ fails perfect recovery when $\boldsymbol{y}' \in \{\pm1\}^N$ exists such that $\boldsymbol{y}' \neq \pm\boldsymbol{y}^*$ and $f(\boldsymbol{y}') > f(\boldsymbol{y}^*)$. To prove 9 we show that if $I(a,b,c) < 1$, there exists a data point u with high probability such that the log-likelihood increases when the label of only u is transformed.

We denote the label of only u transformed as $\boldsymbol{y}'$,

$$y'_u = -y^*_u,\ y'_i = y^*_i (i \neq u)$$

and the difference of the log-likelihoods are

$$\begin{aligned}f(\boldsymbol{y}^*) - f(\boldsymbol{y}'_u) &= \log\frac{\mathbb{P}(\boldsymbol{A}|\boldsymbol{y} = \boldsymbol{y}^*)}{\mathbb{P}(\boldsymbol{A}|\boldsymbol{y} = \boldsymbol{y}')} + \log\frac{\mathbb{P}(\boldsymbol{X}|\boldsymbol{y} = \boldsymbol{y}^*)}{\mathbb{P}(\boldsymbol{X}|\boldsymbol{y} = \boldsymbol{y}')} \\ &= \sum_{i\neq u} \log\frac{\mathbb{P}(A_{iu}|y_u = y^*_u, y_i = y^*_i)}{\mathbb{P}(A_{iu}|y_u = -y^*_u, y_i = y^*_i)} + \log\frac{\mathbb{P}(\boldsymbol{x}_u|y_u = y^*_u)}{\mathbb{P}(\boldsymbol{x}_u|y_u = -y^*_u)}.\end{aligned}$$

We evaluate $\mathbb{P}(\exists u\ s.t.\ f(\boldsymbol{y}^*) - f(\boldsymbol{y}') < 0)$ but it is difficult due to the dependence on $f(\boldsymbol{y}^*) - f(\boldsymbol{y}')$ of each u. So we introduce some random variables to remove this dependence.

First, we choose $\delta_N N, \delta_N = (\log N)^{-1}$ elements from $[N]$ and denote the set of these elements as $\mathcal{U}$. The probability of $u \in \mathcal{U}$ does not depend on u. Second, we define random variables with this $\mathcal{U}$

$$W_u := \sum_{i\in[N]\setminus\mathcal{U}} \log \frac{\mathbb{P}(A_{iu}|y_u = y_u^*, y_i = y_i^*)}{\mathbb{P}(A_{iu}|y_u = -y_u^*, y_i = y_i^*)} + \log \frac{\mathbb{P}(\boldsymbol{x}_u|y_u = y_u^*)}{\mathbb{P}(\boldsymbol{x}_u|y_u = -y_u^*)}$$

$$J_u := \sum_{i\in\mathcal{U}\setminus\{u\}} \log \frac{\mathbb{P}(A_{iu}|y_u = y_u^*, y_i = y_i^*)}{\mathbb{P}(A_{iu}|y_u = -y_u^*, y_i = y_i^*)}, \quad J := \max_{u\in\mathcal{U}} J_u.$$

It is easy to show that $f(\boldsymbol{y}^*) - f(\boldsymbol{y}'_u) = W_u + J_u$ and for each $u, u' \in \mathcal{U}$, W_u and $W_{u'}$ are independent and W_u and J are so on. We decompose the event $\{W_u + J_u < 0\}$ as $\{W_u \le -\zeta_N q_N \cap J_u \le \zeta_N q_N\}$, $\zeta_N = (\log\log\log N)^{-1}$ and

$$\begin{aligned}
\mathbb{P}_{\text{fail}} &\ge \mathbb{P}(\exists u \; s.t. \; f(\boldsymbol{y}^*) - f(\boldsymbol{y}') < 0) = \mathbb{P}\Big(\bigcup_{u\in[N]} \{W_u + J_u < 0\}\Big) \\
&\ge \mathbb{P}\Big(\bigcup_{u\in\mathcal{U}} \{W_u + J_u < 0\}\Big) \ge \mathbb{P}\Big(\bigcup_{u\in\mathcal{U}} \{W_u \le -\zeta_N q_N \cap J_u \le \zeta_N q_N\}\Big) \\
&\ge \mathbb{P}\Big(\{J \le \zeta_N q_N\} \cap \bigcup_{u\in\mathcal{U}} \{W_u \le -\zeta_N q_N\}\Big) \\
&= \mathbb{P}(\{J \le \zeta_N q_N\}) \cdot \mathbb{P}\Big(\bigcup_{u\in\mathcal{U}} \{W_u \le -\zeta_N q_N\}\Big)
\end{aligned}$$

We use the following lemma on these probabilities:

Lemma 13 *Under assumption 1,*

(i) We denote $\zeta'_N = (\log\log N)^{-1}$. There is a constant C that satisfies

$$\mathbb{P}(\{J \le \zeta_N q_N\}) \ge 1 - \frac{C\zeta'_N}{\zeta_N}(1 + o(1)) \, .$$

(ii) For any constant $\tilde{\delta} > 0$

$$\mathbb{P}\Big(\bigcup_{u\in\mathcal{U}} \{W_u \le -\zeta_N q_N\}\Big) \ge 1 - \exp(-\delta_N N^{1-I(a,b,c)-\tilde{\delta}})$$

Therefore, if $I(a,b,c) = 1 - \epsilon < 1$, we choose $\tilde{\delta} < \epsilon$ and

$$\begin{aligned}
\mathbb{P}_{\text{fail}} &\ge \mathbb{P}(\{J \le \zeta_N q_N\}) \cdot \mathbb{P}\Big(\bigcup_{u\in\mathcal{U}} \{W_u \le -\zeta_N q_N\}\Big) \\
&\ge (1 - \frac{C\zeta'_N}{\zeta_N}(1 + o(1))) \cdot (1 - \exp(-\delta_N N^{1-I(a,b,c)-\tilde{\delta}})) \\
&= (1 - \frac{C\zeta'_N}{\zeta_N}(1 + o(1))) \cdot (1 - \exp(-\delta_N N^{\epsilon-\tilde{\delta}})) \underset{N\to\infty}{\to} 1
\end{aligned}$$

so we conclude that the MLE fails to perfect recovery.

D.1. Proof of Lemma 13

Proof [proof of (i)] In this proof, we assume $a > b$ but if $a < b$ the same result will be derived.

$$\begin{aligned}\mathbb{P}(\{J \le \zeta_N q_N\}) &= 1 - \mathbb{P}(\{J > \zeta_N q_N\}) \\ &\ge 1 - \mathbb{P}(\{|J| > \zeta_N q_N\}) \\ &\ge 1 - \frac{\mathbb{E}_{\boldsymbol{y}^*,\mathcal{U},\boldsymbol{A}}[|J|]}{\zeta_N q_N}\end{aligned}$$

In the last line, we use Markov's inequality. If we prove

$$\mathbb{E}_{\boldsymbol{y}^*,\mathcal{U},\boldsymbol{A}}[|J|] \le C\zeta'_N q_N(1+o(1))$$

then this proof will be completed. So we now prove this upper bound.

For any real number $t > 0$,

$$\begin{aligned}\mathbb{E}_{\boldsymbol{y}^*,\mathcal{U},\boldsymbol{A}}[t|J|] &\le \mathbb{E}_{\boldsymbol{y}^*,\mathcal{U},\boldsymbol{A}}\left[t\max_{u\in\mathcal{U}}|J_u|\right] \\ &= \mathbb{E}_{\boldsymbol{y}^*,\mathcal{U},\boldsymbol{A}}\left[\log\exp(t\max_{u\in\mathcal{U}}|J_u|)\right] \\ &\le \log\mathbb{E}_{\boldsymbol{y}^*,\mathcal{U},\boldsymbol{A}}\left[\exp(t\max_{u\in\mathcal{U}}|J_u|)\right] \\ &\le \log\mathbb{E}_{\boldsymbol{y}^*,\mathcal{U},\boldsymbol{A}}\left[\sum_{u\in\mathcal{U}}e^{t|J_u|}\right] = \log\mathbb{E}_{\boldsymbol{y}^*,\mathcal{U}}\left[\sum_{u\in\mathcal{U}}\mathbb{E}_{\boldsymbol{A}}\left[e^{t|J_u|}\middle|\,\boldsymbol{y}^*,\mathcal{U}\right]\right].\end{aligned}$$

We estimate $\mathbb{E}_{\boldsymbol{A}}\left[e^{t|J_u|}\middle|\,\boldsymbol{y}^*,\mathcal{U}\right]$. We denote $n_\pm = \#\{u\in\mathcal{U}|y^*_u = \pm 1\}$ and when $y^*_u = +1$ from the definition of J_u and the distribution of $\boldsymbol{A}$ on XOR-CSBM,

$$\begin{aligned}\mathbb{E}_{\boldsymbol{A}}\left[e^{t|J_u|}\middle|\,\boldsymbol{y}^*,\mathcal{U}\right] &\le \left(\alpha\exp\left(t\left|\log\frac{\alpha}{\beta}\right|\right)+(1-\alpha)\exp\left(t\left|\log\frac{1-\alpha}{1-\beta}\right|\right)\right)^{n_+-1} \\ &\quad\times\left(\beta\exp\left(t\left|\log\frac{\beta}{\alpha}\right|\right)+(1-\beta)\exp\left(t\left|\log\frac{1-\beta}{1-\alpha}\right|\right)\right)^{n_-} \\ &= \left(1+\frac{q_N}{N}\left(a\left(\frac{a}{b}\right)^t - t(a-b)+o(1)\right)\right)^{n_+-1} \\ &\quad\times\left(1+\frac{q_N}{N}\left(b\left(\frac{a}{b}\right)^t - t(a-b)+o(1)\right)\right)^{n_-} \\ &\le \exp\left(\delta_N q_N\left(a\left(\frac{a}{b}\right)^t - t(a-b)+o(1)\right)\right).\end{aligned}$$

In the last inequality, we use $1+x \le e^x$ and $n_+ + n_- - 1 = \delta_N N - 1 = \delta_N N(1+o(1))$. The same upper bound is derived when $y_u^* = -1$, and this upper bound does not depend on $\boldsymbol{y}^*$ and $\mathcal{U}$ so

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{y}^*,\mathcal{U},\boldsymbol{A}}[t|J|] &\le \log \delta_N N \exp\left(\delta_N q_N \left(a\left(\frac{a}{b}\right)^t - t(a-b) + o(1)\right)\right) \\
&= q_N\left(1 + o(1) + \delta_N\left(a\left(\frac{a}{b}\right)^t - t(a-b) + o(1)\right)\right) \\
\mathbb{E}_{\boldsymbol{y}^*,\mathcal{U},\boldsymbol{A}}[|J|] &\le q_N\left(\frac{1+o(1)}{t} + \delta_N\left(\frac{a}{t}\left(\frac{a}{b}\right)^t - (a-b) + o(1)\right)\right)
\end{aligned}$$

When we choose $t = (\log_{a/b} \log N)^{-1} = \log a/b / \log\log N$,

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{y}^*,\mathcal{U},\boldsymbol{A}}[|J|] &= q_N \zeta_N' \left(\frac{1+o(1)}{\log a/b} + \frac{a}{\log a/b} - \delta_N((a-b)+o(1))\right) \\
&= C q_N \zeta_N' (1+o(1))\,.
\end{aligned}$$

■

Proof [proof of (ii)] Because $\zeta_N \to 0$ when $N \to \infty$, for any constants $\delta_1 > 0$ there exists $N_0 > 0$ such that

$$\begin{aligned}
\mathbb{P}\Big(\bigcup_{u\in\mathcal{U}} \{W_u \le -\zeta_N q_N\}\Big) &= 1 - \mathbb{P}\Big(\bigcap_{u\in\mathcal{U}} \{W_u > -\zeta_N q_N\}\Big) \\
&= 1 - \prod_{u\in\mathcal{U}} \mathbb{P}\Big(\{W_u > -\zeta_N q_N\}\Big) \\
&= 1 - \prod_{u\in\mathcal{U}} (1 - \mathbb{P}(\{W_u \le -\zeta_N q_N\})) \\
&\ge 1 - \prod_{u\in\mathcal{U}} (1 - \mathbb{P}(\{W_u \le -\delta_1 q_N\}))
\end{aligned}$$

for all $N > N_0$. We use Theorem H.5 in [1] to estimate $\mathbb{P}(\{W_u \le -\delta_1 q_N\})$, we calculate the moment generating function of W_u.

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{y}^*,\mathcal{U},\boldsymbol{X},\boldsymbol{A}}[e^{tW_u}] &= \mathbb{E}_{\boldsymbol{y}^*,\mathcal{U}}\left[\mathbb{E}_{\boldsymbol{A}}\left[\exp\left(t\sum_{i\in[N]\setminus\mathcal{U}} \log\frac{\mathbb{P}(A_{iu}|y_u = y_u^*, y_i = y_i^*)}{\mathbb{P}(A_{iu}|y_u = -y_u^*, y_i = y_i^*)}\right)\middle|\boldsymbol{y}^*,\mathcal{U}\right]\right. \\
&\quad \left.\times\ \mathbb{E}_{\boldsymbol{X}}\left[\exp\left(t\log\frac{\mathbb{P}(\boldsymbol{x}_u|y_u = y_u^*)}{\mathbb{P}(\boldsymbol{x}_u|y_u = -y_u^*)}\right)\middle|\boldsymbol{y}^*,\mathcal{U}\right]\right] \\
&= \mathbb{E}_{\boldsymbol{y}^*,\mathcal{U}}\left[\mathbb{E}_{\boldsymbol{A}}\left[\exp\left(t\sum_{i\in[N]\setminus\mathcal{U}} \log\frac{\mathbb{P}(A_{iu}|y_u = y_u^*, y_i = y_i^*)}{\mathbb{P}(A_{iu}|y_u = -y_u^*, y_i = y_i^*)}\right)\middle|\boldsymbol{y}^*,\mathcal{U}\right]\right. \\
&\quad \left.\times\ \mathbb{E}_{\boldsymbol{x}_u}\left[\exp\left(t\log\frac{\mathbb{P}(\boldsymbol{x}_u|y_u = y_u^*)}{\mathbb{P}(\boldsymbol{x}_u|y_u = -y_u^*)}\right)\middle|y_u^*\right]\right]
\end{aligned}$$

We use the following lemmas on these expectations:

Lemma 14 *Under assumption 1,*

$$\mathbb{E}_{\boldsymbol{x}_u}\left[\exp\left(t\log\frac{\mathbb{P}(\boldsymbol{x}_u|y_u=y_u^*)}{\mathbb{P}(\boldsymbol{x}_u|y_u=-y_u^*)}\right)\middle|y_u^*\right] = 2\Phi(-t\theta)^2[\exp(t(t+1)\theta^2)\Phi((t+1)\theta) + \exp(t(t-1)\theta^2)\Phi((t-1)\theta)]^2 + o(1).$$

Lemma 15 *Under assumption 1,*

$$\mathbb{E}_{\boldsymbol{y}^*,\mathcal{U}}\left[\mathbb{E}_{\boldsymbol{A}}\left[\exp\left(t\sum_{i\in[N]\setminus\mathcal{U}}\log\frac{\mathbb{P}(A_{iu}|y_u=y_u^*,y_i=y_i^*)}{\mathbb{P}(A_{iu}|y_u=-y_u^*,y_i=y_i^*)}\right)\middle|\boldsymbol{y}^*,\mathcal{U}\right]\right]$$
$$= \left(1-\frac{q_N}{2N}\left(a-a\left(\frac{a}{b}\right)^t+b-b\left(\frac{b}{a}\right)^t+o(1)\right)\right)^{N-\delta_N N}$$

So,

$$\begin{aligned}
-\frac{1}{q_N}\log\mathbb{E}[e^{tW_u}] &= \frac{1}{2}\left(a-a\left(\frac{a}{b}\right)^t+b-b\left(\frac{b}{a}\right)^t\right)-\frac{2}{q_N}\log(\Phi(-t\theta)) \\
&\quad -\frac{2}{q_N}\log(\exp(t(t+1)\theta^2)\Phi((t+1)\theta)+\exp(t(t-1)\theta^2)\Phi((t-1)\theta)) \\
&= \frac{1}{2}\left(a-a\left(\frac{a}{b}\right)^t+b-b\left(\frac{b}{a}\right)^t\right)+\begin{cases}-ct(t+2)+o(1) & t>0\\ -2ct(t+1)+o(1) & -1<t<0\\ -c(t^2-1)+o(1) & t<-1\end{cases} \\
&=: I(t,a,b,c)
\end{aligned}$$

Here we used $x\to-\infty$, $\log\Phi(x)=-\frac{x^2}{2}+O(1)$. Since both of these two terms take their supremum at $t=-\frac{1}{2}$,

$$\sup_{t\in\mathbb{R}}[I(t,a,b,c)]] = \frac{(\sqrt{a}-\sqrt{b})^2+c}{2}+o(1) = I(a,b,c)+o(1)\,.$$

Since $I(t,a,b,c)$ is a convex function of t we can apply Theorem H.5 in [1] to $\mathbb{P}(\{W_u\le-\delta_1 q_N\})$, then

$$\lim_{N\to\infty}\frac{1}{q_N}\log\mathbb{P}(\{W_u\le-\delta_1 q_N\}) = -\sup_{t\in\mathbb{R}}(-\delta_1 t+I(t,a,b,c))\,.$$

This equation is satisfied for all $\delta_1>0$ and the right hand side will be $-\sup_{t\in\mathbb{R}}I(t,a,b,c) = -I(a,b,c)$ when $\delta_1\to0$, so for any $\tilde{\delta}>0$ there exists $N_1>0$ such that

$$\frac{1}{q_N}\log\mathbb{P}(\{W_u\le-\delta_1 q_N\})\ge-\tilde{\delta}-\sup_{t\in\mathbb{R}}I(t,a,b,c) = -\tilde{\delta}-I(a,b,c)$$

for all $N>N_1$. Therefore, for all $N>\max(N_0,N_1)$

$$\begin{aligned}
\mathbb{P}\left(\bigcup_{u\in\mathcal{U}}\{W_u\le-\zeta_N q_N\}\right) &\ge 1-\prod_{u\in\mathcal{U}}\left(1-\exp(-q_N(I(a,b,c)+\tilde{\delta}))\right) \\
&= 1-\left(1-N^{-(I(a,b,c)-\tilde{\delta})}\right)^{\delta_N N} \\
&\ge 1-\exp\left(-\delta_N N^{1-I(a,b,c)-\tilde{\delta})}\right)
\end{aligned}$$

then this proof is completed. ■

D.2. Proof of Lemma 15

Proof For given $\boldsymbol{y}^*, \mathcal{U}$, we denote $n_\pm = \#\{i \in [N] \setminus (\mathcal{U} \cup u) | y_i^* y_u^* = \pm 1\}$.

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{A}}\left[\exp\left(t\sum_{i\in[N]\setminus\mathcal{U}}\log\frac{\mathbb{P}(A_{iu}|y_u=y_u^*,y_i=y_i^*)}{\mathbb{P}(A_{iu}|y_u=-y_u^*,y_i=y_i^*)}\right)\middle|\boldsymbol{y}^*,\mathcal{U}\right]\\
=&\left(\mathbb{E}_{A_{iu}\sim\mathrm{Ber}(\alpha)}\left[\left(\frac{\mathbb{P}(A_{iu}|y_i=y_u)}{\mathbb{P}(A_{iu}|y_i\neq y_u)}\right)^t\right]\right)^{n_+}\left(\mathbb{E}_{A_{iu}\sim\mathrm{Ber}(\beta)}\left[\left(\frac{\mathbb{P}(A_{iu}|y_i\neq y_u)}{\mathbb{P}(A_{iu}|y_i=y_u)}\right)^t\right]\right)^{n_-}\\
=&\left(\alpha\left(\frac{\alpha}{\beta}\right)^t+(1-\alpha)\left(\frac{1-\alpha}{1-\beta}\right)^t\right)^{n_+}\left(\beta\left(\frac{\beta}{\alpha}\right)^t+(1-\beta)\left(\frac{1-\beta}{1-\alpha}\right)^t\right)^{n_-}
\end{aligned}$$

Because of the distribution of $\boldsymbol{y}^*, \mathcal{U}$, $\mathbb{P}(n_+ = k) = \frac{1}{2^{(N-\delta_N N)}}\binom{N-\delta_N N}{k}$. Therefore,

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{y}^*,\mathcal{U}}\left[\mathbb{E}_{\boldsymbol{A}}\left[\exp\left(t\sum_{i\in[N]\setminus\mathcal{U}}\log\frac{\mathbb{P}(A_{iu}|y_u=y_u^*,y_i=y_i^*)}{\mathbb{P}(A_{iu}|y_u=-y_u^*,y_i=y_i^*)}\right)\middle|\boldsymbol{y}^*,\mathcal{U}\right]\right]\\
=&\frac{1}{2^{N-\delta_N N}}\sum_{n_+=0}^{n}\binom{N-\delta_N N}{n_+}\\
&\qquad\times\left(\alpha\left(\frac{\alpha}{\beta}\right)^t+(1-\alpha)\left(\frac{1-\alpha}{1-\beta}\right)^t\right)^{n_+}\left(\beta\left(\frac{\beta}{\alpha}\right)^t+(1-\beta)\left(\frac{1-\beta}{1-\alpha}\right)^t\right)^{n_-}\\
=&\left(\frac{\alpha}{2}\left(\frac{\alpha}{\beta}\right)^t+\frac{1-\alpha}{2}\left(\frac{1-\alpha}{1-\beta}\right)^t+\frac{\beta}{2}\left(\frac{\beta}{\alpha}\right)^t+\frac{1-\beta}{2}\left(\frac{1-\beta}{1-\alpha}\right)^t\right)^{N-\delta_N N}\\
=&\left(1+\frac{q_N}{2N}\left(a\left(\frac{a}{b}\right)^t-a-t(a-b)+b\left(\frac{b}{a}\right)^t-b-t(b-a)+o(1)\right)\right)^{N-\delta_N N}\\
=&\left(1-\frac{q_N}{2N}\left(a-a\left(\frac{a}{b}\right)^t+b-b\left(\frac{b}{a}\right)^t+o(1)\right)\right)^{N-\delta_N N}
\end{aligned}$$

and the proof is completed. ■

D.3. Proof of Lemma 14

Proof Using the distribution of $\boldsymbol{X}$ in XOR-CSBM,

$$\begin{aligned}
\log\frac{\mathbb{P}(\boldsymbol{x}_u|y_u=y_u^*)}{\mathbb{P}(\boldsymbol{x}_u|y_u=-y_u^*)}&=\log\frac{\exp(-\|\boldsymbol{x}_u-\theta\boldsymbol{\mu}_{y_u^*}\|^2/2)+\exp(-\|\boldsymbol{x}_u+\theta\boldsymbol{\mu}_{y_u^*}\|^2/2)}{\exp(-\|\boldsymbol{x}_u-\theta\boldsymbol{\mu}_{-y_u^*}\|^2/2)+\exp(-\|\boldsymbol{x}_u+\theta\boldsymbol{\mu}_{-y_u^*}\|^2/2)}\\
&=\log\frac{\cosh\theta\boldsymbol{x}_u^T\boldsymbol{\mu}_{y_u^*}}{\cosh\theta\boldsymbol{x}_u^T\boldsymbol{\mu}_{-y_u^*}}
\end{aligned}$$

$$\mathbb{E}_{\boldsymbol{X}}\left[\exp\left(t\log\frac{\mathbb{P}(\boldsymbol{x}_u|y_u=y_u^*)}{\mathbb{P}(\boldsymbol{x}_u|y_u=-y_u^*)}\right)\middle|\boldsymbol{y}^*\right]=\mathbb{E}_{\boldsymbol{x}_u}\left[\left(\frac{\cosh\theta\boldsymbol{x}_u^T\boldsymbol{\mu}_{y_u^*}}{\cosh\theta\boldsymbol{x}_u^T\boldsymbol{\mu}_{-y_u^*}}\right)^t\middle|\boldsymbol{y}^*\right]$$

Because $\boldsymbol{x}_u\sim\frac{1}{2}\mathcal{N}(\theta\boldsymbol{\mu}_{y_u^*},\boldsymbol{I}_d)+\frac{1}{2}\mathcal{N}(-\theta\boldsymbol{\mu}_{y_u^*},\boldsymbol{I}_d)$ the expectation is

$$\begin{aligned}\mathbb{E}_{\boldsymbol{x}_u}\left[\left(\frac{\cosh\theta\boldsymbol{x}_u^T\boldsymbol{\mu}_{y_u^*}}{\cosh\theta\boldsymbol{x}_u^T\boldsymbol{\mu}_{-y_u^*}}\right)^t\middle|\boldsymbol{y}^*\right] &= \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)}\left[\left(\frac{\cosh(\theta^2+\theta\boldsymbol{z}^T\boldsymbol{\mu}_{y_u^*})}{\cosh(\theta\boldsymbol{z}^T\boldsymbol{\mu}_{-y_u^*})}\right)^t\right]\\ &\quad+\frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)}\left[\left(\frac{\cosh(-\theta^2+\theta\boldsymbol{z}^T\boldsymbol{\mu}_{y_u^*})}{\cosh(\theta\boldsymbol{z}^T\boldsymbol{\mu}_{-y_u^*})}\right)^t\right]\\ &= \frac{1}{2}\mathbb{E}_{\xi,\zeta\sim\mathcal{N}(0,1)}\left[\left(\frac{\cosh(\theta^2+\theta\xi)}{\cosh(\theta\zeta)}\right)^t\right]\\ &\quad+\frac{1}{2}\mathbb{E}_{\xi,\zeta\sim\mathcal{N}(0,1)}\left[\left(\frac{\cosh(-\theta^2+\theta\xi)}{\cosh(\theta\zeta)}\right)^t\right]\\ &= \mathbb{E}_{\xi\sim\mathcal{N}(0,1)}[(\cosh(\theta^2+\theta\xi))^t]\cdot\mathbb{E}_{\zeta\sim\mathcal{N}(0,1)}[(\cosh\theta\zeta)^{-t}].\end{aligned}$$

From assumption 1 $\theta=\Theta(q_N^{1/2})$ and $\cosh(\theta^2+\theta\xi)\simeq e^{\theta|\theta+\xi|}$, $\cosh\theta\zeta\simeq e^{\theta|\zeta|}$, so

$$\begin{aligned}\mathbb{E}_{\boldsymbol{x}_u}\left[\left(\frac{\cosh\theta\boldsymbol{x}_u^T\boldsymbol{\mu}_{y_u^*}}{\cosh\theta\boldsymbol{x}_u^T\boldsymbol{\mu}_{-y_u^*}}\right)^t\middle|\boldsymbol{y}^*\right] &\simeq \frac{1}{2\pi}\left[\int_0^\infty\exp(-\zeta^2/2-t\theta\zeta)d\zeta\right.\\ &\qquad\left.+\int_{-\infty}^0\exp(-\zeta^2/2+t\theta\zeta)d\zeta\right]\\ &\quad\times\left[\int_{-\theta}^\infty\exp(-\xi^2/2+t\theta^2+t\theta\xi)d\xi\right.\\ &\qquad\left.+\int_{-\infty}^{-\theta}\exp(-\xi^2/2-t\theta^2-t\theta\xi)d\xi\right]\\ &= 2\Phi(-t\theta)\left[e^{t(t+1)\theta^2}\Phi((t+1)\theta)+e^{t(t-1)\theta^2}\Phi((t-1)\theta)\right]\end{aligned}$$

■

Appendix E. Informational Lower Bound of MLP and GCN

We prove that there is a sufficient condition that MLP and GCN predict true labels for all of the test data. We keep in mind the models we learned in Algorithms 1 and 2, but we use neurons and self loops defined in Assumption 2 and Theorem 10.

For an estimator $\hat{\boldsymbol{y}}$ the probability of perfect recovery is

$$\mathbb{P}(\bigcap_{i\in[N]}\{\operatorname{sgn}(y_i)=\operatorname{sgn}(\hat{y}_i)\})=1-\mathbb{P}(\bigcup_{i\in[N]}\{y_i\hat{y}_i<0\})\geq 1-\sum_{i\in[N]}\mathbb{P}(\{y_i\hat{y}_i<0\})\,.$$

Here we used the union bound. If all $\mathbb{P}(\{y_i\hat{y}_i<0\}), i\in[N]$ decrease more quickly than N^{-1} when $N\to\infty$, the right hand side will converge to 1 when $N\to\infty$ and the perfect recovery will be achieved. The following proposition claims that the MLP and GCN satisfy this order condition for the given parameter condition.

Algorithm 1 Learning MLP

Input: $\boldsymbol{\mu}_1, \boldsymbol{\mu}_{-1}, \alpha, \beta, \theta$

Initialization: $\boldsymbol{w}_j \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d), a_j \sim \mathrm{Unif}(\pm 1/\sqrt{K}),\ \forall j \in [K]$

Learning Step:

while $t < T$ **do**

$(\boldsymbol{y}^{(t)}, \boldsymbol{X}^{(t)}, \boldsymbol{A}^{(t)}) \sim \text{XOR-CSBM}(\boldsymbol{\mu}_1, \boldsymbol{\mu}_{-1}, \alpha, \beta, \theta)$

$\boldsymbol{W}^{(t+1)} \leftarrow \boldsymbol{W}^{(t)} - \eta \nabla_{\boldsymbol{W}^{(t)}} L^{\mathrm{MLP}}(\boldsymbol{W}^{(t)}; \boldsymbol{y}^{(t)}, \boldsymbol{X}^{(t)})$

end while

Prediction Step:

$(\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{A}) \sim \text{XOR-CSBM}(\boldsymbol{\mu}_1, \boldsymbol{\mu}_{-1}, \alpha, \beta, \theta)$

Output: $\mathrm{sgn}(f^{\mathrm{MLP}}(\boldsymbol{X}))$

Figure 3: The learning algorithm for MLP

Algorithm 2 Learning GCN

Input: $\boldsymbol{\mu}_1, \boldsymbol{\mu}_{-1}, \alpha, \beta, \theta$

Initialization: $\boldsymbol{w}_j \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d), a_j \sim \mathrm{Unif}(\pm 1/\sqrt{K}),\ \forall j \in [K]$

Learning Step:

while $t < T$ **do**

$(\boldsymbol{y}^{(t)}, \boldsymbol{X}^{(t)}, \boldsymbol{A}^{(t)}) \sim \text{XOR-CSBM}(\boldsymbol{\mu}_1, \boldsymbol{\mu}_{-1}, \alpha, \beta, \theta)$

$\boldsymbol{W}^{(t+1)} \leftarrow \boldsymbol{W}^{(t)} - \eta \nabla_{\boldsymbol{W}^{(t)}} L^{\mathrm{GCN}}(\boldsymbol{W}^{(t)}; \boldsymbol{y}^{(t)}, \boldsymbol{X}^{(t)}, \boldsymbol{A}^{(t)})$

end while

Prediction Step:

$(\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{A}) \sim \text{XOR-CSBM}(\boldsymbol{\mu}_1, \boldsymbol{\mu}_{-1}, \alpha, \beta, \theta)$

$\boldsymbol{A} \leftarrow \boldsymbol{A} + \rho \boldsymbol{I}_N$

Output: $\mathrm{sgn}(f^{\mathrm{GCN}}(\boldsymbol{X}))$

Figure 4: The learning algorithm for GCN

Proposition 16

(i) Let us think of the MLP after learning with algorithm 1and denote its output as $\hat{\boldsymbol{y}}^{\mathrm{MLP}}$*. Under assumptions 1 and 2*

$$\lim_{N\to\infty}\frac{1}{q_N}\log\mathbb{P}(\{y_i\hat{y}_i^{\mathrm{MLP}}<0\})=-\frac{c}{2}\,.$$

(ii) Let us think of the GCN after learning with algorithm 2 with the self loop $\rho=\frac{2c}{\log(a/b)}q_N$ *and denote its output as* $\hat{\boldsymbol{y}}^{\mathrm{GCN}}$*. Under assumptions 1 and 2*

$$\lim_{N\to\infty}\frac{1}{q_N}\log\mathbb{P}(\{y_i\hat{y}_i^{\mathrm{GCN}}<0\})=-I(a,b,c)\,.$$

Roughly speaking, this proposition argues that $\mathbb{P}(\{y_i\hat{y}_i^{\mathrm{MLP}}<0\})=\exp(-q_N\cdot\frac{c}{2})=N^{-\frac{c}{2}}$ and $\mathbb{P}(\{y_i\hat{y}_i^{\mathrm{GCN}}<0\})=\exp(-q_N\cdot I(a,b,c))=N^{-I(a,b,c)}$ so the MLP achieves perfect recovery when $\frac{c}{2}>1$ and the GCN does when $I(a,b,c)>1$.

E.1. Proof of Proposition 16

Proof [proof of (i)] We denote $h=y_1\hat{y}_1^{\mathrm{MLP}}$ for simplification. To use Theorem H.5 in [1], we calculate this moment generating function

$$\mathbb{E}[\exp(th)]=\mathbb{E}_{\boldsymbol{y},\boldsymbol{X},\boldsymbol{W},\boldsymbol{a}}[\exp(th)],\ h=y_1\sum_{k=1}^{K}\sigma(\boldsymbol{x}_1^T\boldsymbol{w}_k)a_k\,.$$

Under the assumption of $(\boldsymbol{w}_k,a_k)$, the expectation of $\boldsymbol{W},\boldsymbol{a}$ becomes

$$\mathbb{E}_{\boldsymbol{W},\boldsymbol{a}}[\exp(th)] \;=\; \exp\left(\frac{y_1t\sqrt{K}w_{sig}}{4}\left(|\boldsymbol{x}_1^T\boldsymbol{\mu}_1|-|\boldsymbol{x}_1^T\boldsymbol{\mu}_{-1}|\right)\right)$$

Here we used $\sigma(x)+\sigma(-x)=|x|$. We don't care the scale of t so we rewrite $\frac{\sqrt{K}w_{sig}}{4}t$ as t, and the expectation of remains is

$$\begin{aligned}
\mathbb{E}[\exp(th)] &= \frac{1}{4}\left(\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\exp\left(t(|\theta+\boldsymbol{z}^T\boldsymbol{\mu}_1|-|\boldsymbol{z}^T\boldsymbol{\mu}_{-1}|)\right)\right]\right.\\
&= +\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\exp\left(t(|-\theta+\boldsymbol{z}^T\boldsymbol{\mu}_1|-|\boldsymbol{z}^T\boldsymbol{\mu}_{-1}|)\right)\right]\\
&= +\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\exp\left(-t(|\boldsymbol{z}^T\boldsymbol{\mu}_1|-|\theta+\boldsymbol{z}^T\boldsymbol{\mu}_{-1}|)\right)\right]\\
&= \left.\frac{1}{4}\,\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\exp\left(-t(|\boldsymbol{z}^T\boldsymbol{\mu}_1|-|-\theta+\boldsymbol{z}^T\boldsymbol{\mu}_{-1}|)\right)\right]\right)\\
&= \mathbb{E}_{z\sim\mathcal{N}(0,1)}\left[e^{t|\theta+z|}\right]\cdot\mathbb{E}_{z\sim\mathcal{N}(0,1)}\left[e^{-t|z|}\right]
\end{aligned}$$

We use the following formula for a random variable $z\sim\mathcal{N}(0,1)$ and any constants C,D

$$\mathbb{E}_{z\sim\mathcal{N}(0,1)}[e^{C|z+D|}]=e^{\frac{C^2}{2}}(e^{CD}\Phi(C+D)+e^{-CD}\Phi(C-D))$$

so, we rescaled t as $x = t/\sqrt{q_N}$ and

$$\begin{aligned}
\mathbb{E}[\exp(th)] &= 2e^{t^2}\Phi(-t)(e^{t\theta}\Phi(t+\theta) + e^{-t\theta}\Phi(t-\theta)) \\
-\frac{1}{q_N}\log\mathbb{E}[\exp(th)] &= -x^2 - \frac{1}{q_N}\log\Phi(-x\sqrt{q_N}) + o(1) \\
&\quad -\frac{1}{q_N}\log(e^{x\sqrt{c}q_N}\Phi((x+\sqrt{c})\sqrt{q_N}) + e^{-x\sqrt{c}q_N}\Phi((x-\sqrt{c})\sqrt{q_N}))) \\
&= \begin{cases} -\frac{x^2}{2} - \sqrt{c}x + o(1) & x > 0 \\ -x^2 - \sqrt{c}x + o(1) & -\sqrt{c} < x < 0 \\ -\frac{x^2}{2} + \frac{c}{2} + o(1) & t < -\sqrt{c} \end{cases} .
\end{aligned}$$

Here we used for $x \ll -1,\ \log\Phi(x) = -\frac{x^2}{2} + O(1)$. Since this function is convex and its supreme is $-\frac{c}{2}$, the proposition is proved by Theorem H.5 in [1]. ■

Proof [proof of (ii)] We denote $h = y_i\hat{y}_i^{\mathrm{GCN}}$ for simplification. The key idea of this proof is the same as the part of (i) and we calculate this moment generating function

$$\mathbb{E}[\exp(th)] = \mathbb{E}_{\boldsymbol{y},\boldsymbol{X},\boldsymbol{A},\boldsymbol{W},\boldsymbol{a}}[\exp(th)],\ h = \frac{y_1}{D}\sum_{j=1}^{N}\sum_{k=1}^{K} A_{1j}\sigma(\boldsymbol{x}_j^T\boldsymbol{w}_k)a_k\ .$$

With the same calculation as part (i), the expectation of $\boldsymbol{W}, \boldsymbol{a}$ will be

$$\mathbb{E}_{\boldsymbol{W},\boldsymbol{a}}[\exp(th)] = \exp\left(\frac{y_1 t\sqrt{K}w_{sig}}{4D}\sum_{j=1}^{N} A_{1j}(|\boldsymbol{x}_j^T\boldsymbol{\mu}_1| - |\boldsymbol{x}_j^T\boldsymbol{\mu}_{-1}|)\right)\ .$$

We rewrite $\frac{\sqrt{K}w_{sig}}{4D}t$ as t, and the expectation of remains is

$$\begin{aligned}
\mathbb{E}[\exp(th)] &= \mathbb{E}_{\boldsymbol{y}}\left[\mathbb{E}_{\boldsymbol{A},\boldsymbol{X}}\left[\exp\left(y_1 t\sum_{j=1}^{N}A_{1j}(|\boldsymbol{x}_j^T\boldsymbol{\mu}_1|-|\boldsymbol{x}_j^T\boldsymbol{\mu}_{-1}|)\right)\middle|\boldsymbol{y}\right]\right]\\
&= \mathbb{E}_{\boldsymbol{y}}\left[\mathbb{E}_{\boldsymbol{x}_1}\left[e^{y_1 t q_N s(|\boldsymbol{x}_j^T\boldsymbol{\mu}_1|-|\boldsymbol{x}_j^T\boldsymbol{\mu}_{-1}|)}\middle|\boldsymbol{y}\right]\right.\\
&\quad\left.\times\prod_{j=2}^{N}\mathbb{E}_{A_{1j},\boldsymbol{x}_j}\left[e^{y_1 tA_{1j}(|\boldsymbol{x}_j^T\boldsymbol{\mu}_1|-|\boldsymbol{x}_j^T\boldsymbol{\mu}_{-1}|)}|\boldsymbol{y}\right]\right]\\
&= \frac{1}{2}\left(\mathbb{E}_{\boldsymbol{z}_1\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)}\left[e^{tq_N s(|\theta+\boldsymbol{z}_1^T\boldsymbol{\mu}_1|-|\boldsymbol{z}_1^T\boldsymbol{\mu}_{-1}|)}\right]\right.\\
&\quad\times\left[\frac{1}{2}\mathbb{E}_{A_{12},\boldsymbol{z}_2\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)}\left[e^{tA_{1j}(|\theta+\boldsymbol{z}_2^T\boldsymbol{\mu}_1|-|\boldsymbol{z}_2^T\boldsymbol{\mu}_{-1}|)}\,|y_1=1,y_2=1\right]\right.\\
&\quad\left.+\frac{1}{2}\mathbb{E}_{A_{12},\boldsymbol{z}_2\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)}\left[e^{tA_{1j}(|\boldsymbol{z}_2^T\boldsymbol{\mu}_1|-|\theta+\boldsymbol{z}_2^T\boldsymbol{\mu}_{-1}|)}\,|y_1=1,y_2=-1\right]\right]^{N-1}\\
&\quad+\mathbb{E}_{\boldsymbol{z}_1\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)}\left[e^{-tq_N s(|\boldsymbol{z}_1^T\boldsymbol{\mu}_1|-|\theta+\boldsymbol{z}_1^T\boldsymbol{\mu}_{-1}|}\right]\\
&\quad\times\left[\frac{1}{2}\mathbb{E}_{A_{12},\boldsymbol{z}_2\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)}\left[e^{-tA_{1j}|\theta+\boldsymbol{z}_2^T\boldsymbol{\mu}_1|-|\boldsymbol{z}_2^T\boldsymbol{\mu}_{-1}|)}|y_1=-1,y_2=1\right]\right.\\
&\quad\left.\left.+\frac{1}{2}\mathbb{E}_{A_{12},\boldsymbol{z}_2\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)}\left[e^{-tA_{1j}(|\boldsymbol{z}_2^T\boldsymbol{\mu}_1|-|\theta+\boldsymbol{z}_2^T\boldsymbol{\mu}_{-1}|)}|y_1=-1,y_2=-1\right]\right]^{N-1}\right)\\
&= \mathbb{E}_{\boldsymbol{z}_1\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)}\left[e^{tq_N s(|\theta+\boldsymbol{z}_1^T\boldsymbol{\mu}_1|-|\boldsymbol{z}_1^T\boldsymbol{\mu}_{-1}|)}\right]\\
&\quad\times\left[1-\frac{\alpha+\beta}{2}+\frac{\alpha}{2}\mathbb{E}_{\boldsymbol{z}_2\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)}\left[e^{t(|\theta+\boldsymbol{z}_2^T\boldsymbol{\mu}_1|-|\boldsymbol{z}_2^T\boldsymbol{\mu}_{-1}|)}\right]\right.\\
&\quad\left.+\frac{\beta}{2}\mathbb{E}_{\boldsymbol{z}_2\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)}\left[e^{-t(|\theta+\boldsymbol{z}_2^T\boldsymbol{\mu}_1|-|\boldsymbol{z}_2^T\boldsymbol{\mu}_{-1}|)}\right]\right]^{N-1}.
\end{aligned}$$

Because $\boldsymbol{z}_1,\boldsymbol{z}_2\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)$, $\boldsymbol{z}_1^T\boldsymbol{\mu}_1$, $\boldsymbol{z}_1^T\boldsymbol{\mu}_{-1}$, $\boldsymbol{z}_2^T\boldsymbol{\mu}_1$ and $\boldsymbol{z}_2^T\boldsymbol{\mu}_{-1}$ are generated independently with $\mathcal{N}(0,1)$. We use the same formula as the part (i) and the expectation becomes

$$\begin{aligned}
\mathbb{E}[\exp(th)] &= 2e^{t^2s^2q_N^2}\Phi(-tsq_N)(e^{ts\theta q_N}\Phi(tsq_N+\theta)+e^{-ts\theta q_N}\Phi(tsq_N-\theta))\\
&\quad\times\left[1-\frac{\alpha+\beta}{2}+\alpha e^{t^2}\Phi(-t)(e^{t\theta}\Phi(t+\theta)+e^{-t\theta}\Phi(t-\theta))\right.\\
&\quad\left.+\beta\Phi(-t)e^{t^2}(e^{-t\theta}\Phi(-t+\theta)+e^{t\theta}\Phi(-t-\theta))\right]^{N-1}
\end{aligned}$$

$$
\begin{aligned}
\frac{-1}{q_N}\log\mathbb{E}[\exp(th)] &= -t^2s^2q_N - \frac{1}{q_N}\log\Phi(-tsq_N) \\
&\quad -\frac{1}{q_N}\log(e^{ts\theta q_N}\Phi(tsq_N+\theta) + e^{-ts\theta q_N}\Phi(tsq_N-\theta)) \\
&\quad +\frac{a+b}{2} - ae^{t^2}\Phi(-t)(e^{t\theta}\Phi(t+\theta) + e^{-t\theta}\Phi(t-\theta)) \\
&\quad -b\Phi(-t)e^{t^2}(e^{-t\theta}\Phi(-t+\theta) + e^{t\theta}\Phi(-t-\theta)) + o(1) \\
&= -x^2s^2 - \frac{1}{q_N}\log\Phi(-xs\sqrt{q_N}) \\
&\quad -\frac{1}{q_N}\log(e^{xs\sqrt{c}q_N}\Phi((xs+\sqrt{c})\sqrt{q_N}) + e^{-xs\sqrt{c}q_N}\Phi((xs-\sqrt{c})\sqrt{q_N})) \\
&\quad +\frac{a+b}{2} - e^{\frac{x^2}{q_N}}\Phi\left(-\frac{x}{\sqrt{q_N}}\right) \\
&\quad \times\left[a\left(e^{x\sqrt{c}}\Phi\left(\frac{x}{\sqrt{q_N}}+\sqrt{cq_N}\right) + e^{-x\sqrt{c}}\Phi\left(\frac{x}{\sqrt{q_N}}-\sqrt{cq_N}\right)\right)\right. \\
&\quad \left. +b\left(e^{-x\sqrt{c}}\Phi\left(\frac{-x}{\sqrt{q_N}}+\sqrt{cq_N}\right) + e^{x\sqrt{c}}\Phi\left(\frac{-x}{\sqrt{q_N}}-\sqrt{cq_N}\right)\right)\right] \\
&= I_1(x) + I_2(x)
\end{aligned}
$$

Here we define a new variable $x = t/\sqrt{q_N}$ and

$$
\begin{aligned}
I_1(x) &:= -x^2s^2 - \frac{1}{q_N}\log\Phi(-xs\sqrt{q_N}) \\
&\quad -\frac{1}{q_N}\log(e^{xs\sqrt{c}q_N}\Phi((xs+\sqrt{c})\sqrt{q_N}) + e^{-xs\sqrt{c}q_N}\Phi((xs-\sqrt{c})\sqrt{q_N})) \\
I_2(x) &:= \frac{a+b}{2} - e^{\frac{x^2}{q_N}}\Phi\left(-\frac{x}{\sqrt{q_N}}\right) \\
&\quad \times\left[a\left(e^{x\sqrt{c}}\Phi\left(\frac{x}{\sqrt{q_N}}+\sqrt{cq_N}\right) + e^{-x\sqrt{c}}\Phi\left(\frac{x}{\sqrt{q_N}}-\sqrt{cq_N}\right)\right)\right. \\
&\quad \left. +b\left(e^{-x\sqrt{c}}\Phi\left(\frac{-x}{\sqrt{q_N}}+\sqrt{cq_N}\right) + e^{x\sqrt{c}}\Phi\left(\frac{-x}{\sqrt{q_N}}-\sqrt{cq_N}\right)\right)\right]
\end{aligned}
$$

With $q_N, \theta \gg 1$ from the assumption 1 and for $x \ll -1,\ \log\Phi(x) = -\frac{x^2}{2} + O(1)$, I_1 and I_2 are

$$
\begin{aligned}
I_1(x) &= \begin{cases} -\frac{x^2s^2}{2} - xs\sqrt{c} + o(1) & x \geq 0 \\ -x^2s^2 - xs\sqrt{c} + o(1) & -\frac{\sqrt{c}}{s} \leq x \leq 0 \\ -\frac{x^2s^2}{2} + \frac{c}{2} + o(1) & x \leq -\frac{\sqrt{c}}{s} \end{cases} \\
I_2(x) &= \frac{a+b}{2} - \frac{ae^{x\sqrt{c}} + be^{-x\sqrt{c}}}{2} + o(1)
\end{aligned}
$$

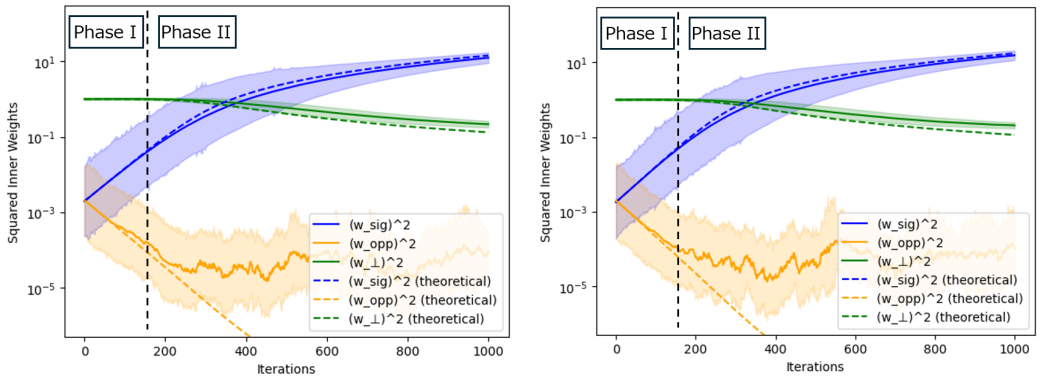
Therefore, the upper bound of $-\frac{1}{q_N}\log\mathbb{E}[\exp(th)]$ is $I(a,b,c)=\frac{(\sqrt{a}-\sqrt{b})^2+c}{2}$, which is attained at $x=-\frac{\sqrt{c}}{s}$. Because $I_1(x), I_2(x)$ are convex function, with Theorem H.5 in [1],

$$\lim_{N\to\infty}\frac{1}{q_N}\log\mathbb{P}(\{h<0\})=-\sup_{x\in\mathbb{R}}(I_1(x)+I_2(x))=-I(a,b,c)$$

and this proof is completed. ■

Appendix F. Experiments

Learning Dynamics of GCN Figure 5 shows the results of the neuron dynamics observed in simulated training of the GCN with the offline gradient descent and numerical solutions of (1). From Figure 5, although the solution in (1) differs from the experimental results in the latter part of the experiment, we confirm that there are two phases. In Phase I w_{sig} increases exponentially, w_{opp} decreases exponentially, and $w_\perp$ dose does not change. In Phase II $\boldsymbol{w}_\perp$ begins to decrease, and w_{sig} continues to increase but the rate of increase slows down.

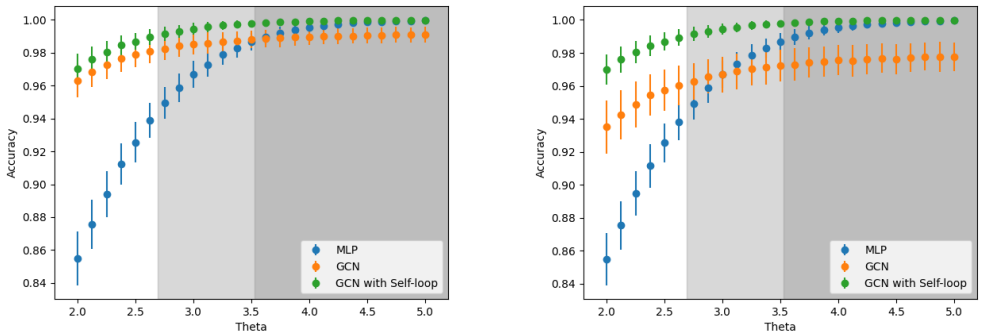


(*a*) homophilic case ($\alpha=0.15,\beta=0.05$) (*b*) heterophilic case ($\alpha=0.05,\beta=0.15$)

Figure 5: The dynamics of $\boldsymbol{w}_{sig},\boldsymbol{w}_{opp},\boldsymbol{w}_\perp$. $N=1000, d=500, K=500, \theta=1.5, \eta=1$. The solid lines are the averages of $|w_{i,sig}|^2, |w_{i,opp}|^2, \|\boldsymbol{w}_{i,\perp}\|^2$ and the dotted lines are the numerical solutions of (1).

Experiments of Perfect Recovery Figure 6 shows the accuracy of the MLP, GCN, GCN with optimal self loops. To see the perfect recovery we used models with parameters of 2. The white areas are where no estimators can achieve perfecta recovery, the light gray areas are where only GCN with optimal self loops achieves perfect recovery, while the dark gray areas are where both GCN with optimal self loops and MLP achieve perfect recovery.

Although perfect recovery has not been achieved because we are considering a finite N, Figure 6 shows that the GCN with self-loops outperforms the MLP. In addition, Figure 6 indicates a regime where the GCN outperforms the MLP even without self-loop optimization. Because perfect recovery is impossible for any estimator there, understanding finer notions of performance, such as prediction accuracy below the perfect recovery threshold, remains an important open problem.



(a) homophilic case ($a = 7, b = 3$) (b) heterophilic case ($a = 3, b = 7$)

Figure 6: The accuracy of MLP, GCN, and GCN with optimal self loop. For $N = 500, d = 200$ and fixed a, b, we calculated the accuracy varying θ. We generated test data 1000 times and calculated the accuracy, and plotted the average and standard deviation. The white areas are where no estimators can achieve perfect recovery ($I(a, b, c) < 1$), the light gray areas are where only GCN with optimal self loops achieves perfect recovery ($I(a, b, c) > 1$ and $\frac{c}{2} < 1$), and the dark gray areas are where both GCN with optimal self loops and MLP achieve perfect recovery ($\frac{c}{2} > 1$).