LEARNING GUARANTEES FOR GRAPH CONVOLUTIONAL NETWORKS ON THE STOCHASTIC BLOCK MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

An abundance of neural network models and algorithms for diverse tasks on graphs have been developed in the past five years. However, very few provable guarantees have been available for the performance of graph neural network models. This state of affairs is in contrast with the steady progress on the theoretical underpinnings of traditional dense and convolutional neural networks. In this paper we present the first provable guarantees for one of the best-studied families of graph neural network models, Graph Convolutional Networks (GCNs), for semi-supervised community detection tasks. We show that with high probability over the initialization and training data, a GCN will efficiently learn to detect communities on graphs drawn from a stochastic block model. Our proof relies on a fine-grained analysis of the training dynamics in order to overcome the complexity of a non-convex optimization landscape with many poorly-performing local minima.

1 INTRODUCTION

There is presently a large gap between what can be accomplished in practice using deep learning, and what can be satisfactorily explained and predicted by the theory of deep learning. Nevertheless, the past several years have seen substantial developments in the theory of deep learning (Ge et al., 2017; Brutzkus & Globerson, 2017; Zhang et al., 2019a; Goel et al., 2020; Chen et al., 2020a).

One factor contributing to the gap between the theory and practice of traditional NNs is that real-world data sets tend to have complex structure that is difficult to capture with formal definitions. For example, popular image classification models are capable of memorizing arbitrary data (Zhang et al., 2016), and yet they exhibit astonishing generalization performance on accurately-labeled natural images. Hence, any rigorous proof of the observed generalization performance of deep learning models on image classification tasks will necessarily require assumptions about the data that are sharp enough to separate random inputs from natural images. Because of the difficulty of giving an adequate characterization of real-world data, much of the recent progress in deep learning theory has instead focused on proving results using very simple (e.g. Gaussian) input distributions or in distribution-free settings (Ge et al., 2017; Brutzkus & Globerson, 2017; Zhang et al., 2019a; Vempala & Wilmes, 2019).

Compared to traditional feed-forward (dense, convolutional, etc.) NNs, the theory of *graph* neural networks (GNNs) is still in its infancy. On the other hand, it appears substantially easier to give plausible descriptions of the combinatorial structure of real-world graph data sets than, e.g., to characterize the distribution of natural images (Drobyshevskiy & Turdakov, 2019). We therefore believe that GNNs offer a natural setting for developing provable guarantees that are able to capture the power of deep learning on real-world datasets. In this paper, we contribute to that goal by giving the first rigorous guarantees of efficient semi-supervised learning of stochastic block models via a GNN.

1.1 GRAPH NEURAL NETWORKS

Many natural datasets for diverse machine learning problems have a graph structure, including social networks, molecular structures, and transit networks. In order to efficiently exploit such combinatorial

structure, a variety of GNN models have been proposed, tuned for different kinds of tasks. A number of taxonomies of GNN models have been proposed (Zhou et al., 2018; Wu et al., 2021); one of the most essential differences between different GNN models is whether they are meant to label the graph as a whole, or to label individual components of the graph, particularly vertices.

From a theoretical perspective, the best understood tasks for GNNs concern labeling the graph as a whole, for example for the task of classifying a graph by its isomorphism type (Sato, 2020). In particular, it has been established that many GNN models are of comparable power to various versions of the Weisfeiler-Leman hierarchy (Xu et al., 2018; Morris et al., 2019).

Some progress has also been made on the theory of GNNs for vertex-labeling tasks. Recent works by Sato et al. describe the representational power of certain GNN models for tasks such as computing minimum vertex covers (2019). Garg et al. also give bounds on the representational power of GNN models, as well as using Rademacher bounds to estimate the generalization ability of GNNs (2020).

Our results concern the task of semi-supervised community detection. In this problem, each vertex belongs to one community, and some subset of the vertices are labeled according to their community membership. The task is to classify the community membership of the remaining vertices. This task has been one of the most intensively studied problems in the GNN literature, but there have not yet been any provable guarantees on the performance of proposed models.

We study (spatial-based) graph convolutional models similar to the GCN model proposed in Kipf & Welling (2017). A single layer of such a model computes weights at each node by aggregating the weights at neighboring nodes and applying an activation function with learned parameters, e.g., a linear map followed by a ReLU. Many variations on this theme, including various sophisticated training regimes, have been proposed (Chen et al., 2017; Gao et al., 2018; Li et al., 2018; Zhang et al., 2019b; Chen et al., 2018), but no provable guarantees have been available for the performance of such models on natural data distributions, until the present work.

2 Main Results

One motivation for GNNs as a target for progress in deep learning theory is that there are well-studied graph distributions that plausibly capture some of the structure of real-world data (Drobyshevskiy & Turdakov, 2019). For example, even fairly simple preferential attachment models plausibly capture some of the essential structure of the web (Kumar et al., 2000). Other graph models naturally capture community structures, the simplest of which is the Stochastic Block Model (SBM) (Holland et al., 1983). A graph is sampled from a SBM by first partitioning vertices into communities (with fixed or random sizes), and then sampling edges independently from Bernoulli distributions whose probabilities depend only on the communities of the endpoints of the edge.

In this paper, we consider the case of an SBM with two blocks, parameterized as SBM(n, p, q) where n is the number of vertices, p is the probability of an intra-community connection, and q is the probability of a cross-community connection. We allow n to vary (but will require it to be sufficiently large), while p and q are of the form $p = \frac{a \log^3 n}{n}$ and $q = \frac{b \log^3 n}{n}$ for some fixed constants $a > b$. The two communities are labeled 0 and 1; we denote the label of vertex x by $\ell(x) \in \{0, 1\}$. In the semi-supervised setting, the community labels of some portion of the labels are revealed. We assume the label of each vertex is revealed independently with probability λ. The input-layer features at a vertex x is $(0, 0)$ if its label is not revealed, $(1, 0)$ if its label is revealed to be 0, and $(0, 1)$ if its label is revealed to be 1.

Assumption 2.1 (Sparse Stochastic Block Model)**.** *The probabilities of intra and cross-community connections are $p = \frac{a \log^3 n}{n}$ and $q = \frac{b \log^3 n}{n}$, where $a > b$ are constants.*

We study the problem of recovering the communities from such graphs using GNN models. Of course, recovering the blocks of an SBM graph has been well-studied and its computational complexity is fully understood in most cases (Abbe & Sandon, 2015; Kawamoto et al., 2019). SBM models are therefore a natural test-case for understanding the power of GNN models for learning community structure, and experimental studies have been done in this setting (Chen et al., 2020b; Yadav et al., 2019), . The contribution here is showing that (multi-layer) GCNs solve the classification problem, which is very much not trivial (it is non-convex, and the training loss curve is empirically non-monotonic).

Our GNN models will be trained on a graph or several graphs generated by the SBM(n, p, q) model, and seek to understand their accuracy on arbitrary SBM(n, p, q) graphs not necessarily in the training set but with the same parameters a, b determining p and q (with n allowed to vary).

In particular, we study spatial-based graph convolutional models along the lines of the Graph Convolutional Networks (GCN) introduced in (Kipf & Welling, 2017). Each layer of the model computes a feature vector at every vertex of an input graph based on features of nearby vertices in the previous layer. A typical layer-wise update rule is of the form

$$X^{(k+1)} = \phi\big(\hat{A}X^{(k)}W^{(k)}\big),$$

where

- $\hat{A}$ is a suitably-normalized adjacency matrix of shape $n \times n$ where n is the number of vertices. Usually $\hat{A}$ includes self-loops.
- $X^{(k)}$ gives the feature vector in the k-th layer at each vertex as a matrix of shape $n \times m_k$, where m_k is the number of features in layer k.
- ϕ is an activation function, such as the ReLU.
- $W^{(k)}$ are the trainable weights in the k-th layer, a matrix of shape $m_k \times m_{k+1}$.

In our version of this model, we define $\hat{A} = \frac{1}{\frac{n}{2}(p+q)}\tilde{A}$, where $\tilde{A} = A + I$, A is the adjacency matrix of a given graph, and I is the identity matrix. For the given SBM(n, p, q), a randomly selected vertex has $\frac{n}{2}(p + q)$ neighbors in expectation, so $\hat{A}$ is obtained by normalizing each row of $A + I$ with the average size of a neighborhood. Since very deep GCN models seem to provide little empirical benefit (Li et al., 2018), we use a single hidden layer with a softmax output layer. Furthermore, we introduce a bias term b at the second layer. So the model has the following form:

$$\begin{aligned} f(X, A) &= \text{softmax}\big(\hat{A}\phi\big(\hat{A}XW^{(0)}\big)W^{(1)} + b\big) \\ &= \text{softmax}\bigg(\frac{4}{n^2(p+q)^2}\tilde{A}\phi\big(\tilde{A}XW^{(0)}\big)W^{(1)} + b\bigg), \end{aligned} \tag{1}$$

where X is the input feature of the graph and $W^{(0)}, W^{(1)}$ and b are trainable parameters. Let h denote the number of hidden features, which equals the number of columns of $W^{(0)}$ and the number of rows of $W^{(1)}$.

The bias term is useful in our analysis because its derivative controls how imbalanced the current loss is between the classes. We consider the cross-entropy loss in training, and have

$$\mathbb{E}[\frac{\partial L}{\partial b_0}] = -\mathbb{E}[\frac{\partial L}{\partial b_1}] = -\frac{1}{2}(\mathbb{E}[Z|\ell(x) = 0] - \mathbb{E}[Z|\ell(x) = 1]),$$

where $Z = \frac{\exp(g_{1-\ell(x)}(x))}{\exp(g_0(x))+\exp(g_1(x))}$. Z can be regarded as a measure of wrong prediction: the numerator is the exponential of the output corresponding to the wrong label and the denominator is a normalizer. It is easy to see that $Z > \frac{1}{2}$ if prediction is wrong; $Z < \frac{1}{2}$ if prediction is correct. When $\big|\mathbb{E}[\frac{\partial L}{\partial b_0}]\big| \approx 0$, the model's loss is balanced in the sense of that $\big|\mathbb{E}[Z|\ell(x) = 0] - \mathbb{E}[Z|\ell(x) = 1]\big| \approx 0$. In order to have balanced performance in every epoch, we train the model through coordinate descent instead of conventional gradient descent. Specifically, in each epoch we first update b_0 and b_1 until $\big|\mathbb{E}[\frac{\partial L}{\partial b_0}]\big|$ is smaller than some threshold. Then we update other parameters.

We define the accuracy of the model as the probability of predicting correctly the label of a single vertex in a randomly generated SBM(n, p, q) graph. We can now state our main result.

Theorem 2.2. *Let $\epsilon > 0$. Given a GCN model as above with $h = O(n)$ hidden features and with parameters initialized independently from $N(0, 1)$, if each training graph is an SBM(n, p, q) with $n \geq \Omega(\frac{1}{\epsilon})^2$ and p and q as described above, and the model is trained according to the above coordinate descent schedule for $k = O(\log\log\frac{1}{\epsilon})$ epochs, then with probability $\geq 1 - O(\frac{1}{h})$, the model achieves accuracy $\geq 1 - 4\epsilon$.*

Remark. The result holds with probability $1 - O(\frac{1}{h})$. Note that $n > \Omega(\frac{1}{\epsilon})^2$ and $h \leq O(n)$, so the probability is indeed also related to n and ϵ. Our analysis produces error terms depending on n and ϵ

which are of lower order than the $O(\frac{1}{h})$ term. We treat λ as constants, so it is omitted in the big O notation. As we shall see in equation (4), bigger λ generates bigger gap, so fewer training epochs are needed to achieve desired accuracy.

We emphasize that the novelty of this theorem is not in learning two-class SBM models as such; this is a long-solved problem. Instead, this is the first proof of efficient learning for a GCN on semi-supervised community detection tasks using a natural family of random graph models.

3 NOTATION

Given a vertex y, denote the row of $\tilde{A}X$ corresponding to y as (t_0^y, t_1^y), so t_0^y and t_1^y give the numbers of neighbors of y (including perhaps y itself) with revealed labels in class 0 and class 1 respectively. Let

$$W^{(0)} = \begin{pmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_h \\ \alpha_1' & \alpha_2' & \cdots & \alpha_h' \end{pmatrix}, \; W^{(1)} = \begin{pmatrix} \beta_1 & \beta_1' \\ \beta_2 & \beta_2' \\ \vdots & \vdots \\ \beta_h & \beta_h' \end{pmatrix}, \; b = \begin{pmatrix} b_0 & b_1 \\ b_0 & b_1 \\ \vdots & \vdots \\ b_0 & b_1 \end{pmatrix}.$$

Then $\alpha_i t_0^y + \alpha_i' t_1^y, 1 \le i \le h$ gives h features of vertex y in the hidden layer. The inner product of the yth row of $\phi(\tilde{A}XW^{(0)})$ and the columns of $W^{(1)}$ gives weighted sums of features of y : $\sum_{i=1}^h \beta_i \phi(\alpha_i t_0^y + \alpha_i' t_1^y)$ and $\sum_{i=1}^h \beta_i' \phi(\alpha_i t_0^y + \alpha_i' t_1^y)$, where ϕ represents the ReLU function.

Given a vertex x, the row of $\hat{A}\phi(\hat{A}XW^{(0)})W^{(1)}$ corresponding to x is denoted by $(f_0(x), f_1(x))$ and is of the form

$$\left(\frac{4}{n^2(p+q)^2} \sum_{y \in G} \mathbb{1}[y \sim x] \sum_{i=1}^h \beta_i \phi(\alpha_i t_0^y + \alpha_i' t_1^y), \frac{4}{n^2(p+q)^2} \sum_{y \in G} \mathbb{1}[y \sim x] \sum_{i=1}^h \beta_i' \phi(\alpha_i t_0^y + \alpha_i' t_1^y) \right), \tag{2}$$

where $\mathbb{1}[y \sim x]$ is equal to 1 if y and x are connected, 0 otherwise. Denote

$$f_0^i(x) := \frac{4\beta_i}{n^2(p+q)^2} \sum_{y \in G} \mathbb{1}[y \sim x] \phi(\alpha_i t_0^y + \alpha_i' t_1^y) \quad f_1^i(x) := \frac{4\beta_i'}{n^2(p+q)^2} \sum_{y \in G} \mathbb{1}[y \sim x] \phi(\alpha_i t_0^y + \alpha_i' t_1^y),$$

so $f_0(x) = \sum_{i=1}^h f_0^i(x)$ and $f_1(x) = \sum_{i=1}^h f_1^i(x)$.

Denote the logits as $(g_0(x), g_1(x))$, where $g_j(x) := f_j(x) + b_j, j = 0, 1$. Denote $\Delta(x) := g_0(x) - g_1(x)$. In order to make correct predictions, we need $\Delta(x) > 0$ when $\ell(x) = 0$ and $\Delta(x) < 0$ when $\ell(x) = 1$.

4 STRUCTURE OF THE PAPER

The performance of the model depends on the concentration and separation of $\Delta(x)$ for $\ell(x) = 0$ and $\ell(x) = 1$ respectively. In Section 5 we show that $\Delta(x)$ is concentrated at one of two values, denoted by μ_0 and μ_1, depending only on whether the label $\ell(x)$ is 0 or 1. The proof depends on different parameter regimes of hidden neurons. In Section 6, we analyze the dynamics of hidden neurons throughout training to show that the concentration and separation improve at a controlled rate. Based on this information, in Section 7 we prove the main theorem. Section 8 shows some experimental results to verify our theory. The paper ends with future directions in Section 9.

5 CONCENTRATION AND SEPARATION OF OUTPUT

The difference of the logits is

$$\Delta(x) = f_0(x) - f_1(x) + b_0 - b_1 = \sum_{i=1}^h \Delta_i(x) + b_0 - b_1,$$

where

$$\Delta_i(x) = f_0^i(x) - f_1^i(x) = \frac{4(\beta_i - \beta_i')}{n^2(p+q)^2} \sum_{y \in G} \mathbb{1}[y \sim x]\phi(\alpha_i t_0^y + \alpha_i' t_1^y).$$

For brevity, we write $\Delta(x)$ as Δ and $\Delta_i(x)$ as Δ_i. In order to estimate Δ, we need to estimate each $\Delta_i, 1 \leq i \leq h$.

Our fine-grained analysis of the dynamics of coordinate descent on GCNs relies on a classification of neurons into three families based on the sign and scale of the parameters: "good type", "bad type" and "harmless type". The names also indicate whether the neuron has positive contribution to the value of $\mu_0 - \mu_1$. We show that "good type" neuron makes positive contribution; the contribution of "bad type" neuron is negative but lower bounded; "harmless type" neuron's contribution is non negative (see Corollary 5.3 and the remark following it). We will specifically describe parameter regime of each type in the following subsections. We analyze the dynamics of these types throughout coordinate descent in the next section. First we give some definitions.

Definition 1. For $1 \leq i \leq h$, we call $(\alpha_i, \alpha_i', \beta_i, \beta_i')$ the **i-th neuron** of the model, where $(\alpha_i, \alpha_i')^\top$ is the i-th column of $W^{(0)}$, (β_i, β_i') is the i-th row of $W^{(1)}$.

Definition 2. We say that the ith neuron is **order-aligned** if $(\alpha_i - \alpha_i')(\beta_i - \beta_i') > 0$, otherwise we say it is **order-misaligned**.

5.1 Classification of neuron parameter regimes

We say the i-th neuron is of **"good type"** if it satisfies either equation G_1 or equation G_2 below. (There is also the symmetric case obtained by switching α_i with α_i' and β_i with β_i'. For brevity, we only consider the cases that $\alpha_i > \alpha_i'$. This applies to the "bad" and "harmless" types below as well). Neurons in this type are order-aligned and both α_i and α_i' are positive or the ratio between α_i and α_i' is large enough.

$$\alpha_i > \alpha_i' > 0 \text{ and } \beta_i > \beta_i' \tag{G_1}$$

$$\alpha_i > 0 > \alpha_i', \left|\frac{\alpha_i}{\alpha_i'}\right| > 1 \text{ and } \beta_i > \beta_i' \tag{G_2}$$

We say the i-th neuron is of **"bad type"** if it satisfies either equation B_1 or equation B_2. Neurons in this type are order-misaligned and α_i, α_i' are either both positive or have the opposite signs.

$$\alpha_i > \alpha_i' > 0 \text{ and } \beta_i < \beta_i' \tag{B_1}$$

$$\alpha_i > 0 > \alpha_i', \left|\frac{\alpha_i}{\alpha_i'}\right| > \frac{q}{p}(1 + \log^{-\frac{1}{3}} n) \text{ and } \beta_i < \beta_i' \tag{B_2}$$

$$\alpha_i > 0 > \alpha_i', \left|\frac{\alpha_i}{\alpha_i'}\right| \leq \frac{q}{p}(1 + \log^{-\frac{1}{3}} n) \tag{B_3}$$

We say that the i-th neuron is of **"harmless type"** if it satisfies either equation H_1 or equation H_2:

$$\alpha_i > 0 > \alpha_i', \left|\frac{\alpha_i}{\alpha_i'}\right| \in (\frac{q}{p}(1 + \log^{-\frac{1}{3}} n), 1] \text{ and } \beta_i > \beta_i' \tag{H_1}$$

$$\alpha_i \leq 0 \text{ and } \alpha_i' \leq 0 \tag{H_2}$$

5.2 Concentration and Separation

Theorem 5.1. *If the i-th neuron is of "good type" satisfying equation G_1 or of "bad type" satisfying equation B_1, then for $\ell(x) = 0$:*

$$\mathbb{P}[\big|\Delta_i - \frac{\lambda(\beta_i - \beta_i')}{(p+q)^2}[(p^2+q^2)\alpha_i + 2pq\alpha_i']\big| \leq (\alpha_i - \alpha_i')(\beta_i - \beta_i')O(\log^{-\frac{1}{2}} n) | \ell(x) = 0] \geq 1 - O(\frac{1}{n^2}),$$

for $\ell(x) = 1$*:*

$$\mathbb{P}[\big|\Delta_i - \frac{\lambda(\beta_i - \beta_i')}{(p+q)^2}[2pq\alpha_i + (p^2+q^2)\alpha_i']\big| \leq (\alpha_i - \alpha_i')(\beta_i - \beta_i')O(\log^{-\frac{1}{2}} n)|\ell(x) = 1] \geq 1 - O\big(\frac{1}{n^2}\big).$$

Similar concentration hold for neurons satisfying equation G_2 *and equation* B_2*, and for neurons of "harmless type."*

We apply the method of bounded differences to show the concentration. The details are shown in the appendix.

Given the concentration of Δ_i for each type of neurons, we estimate the concentration of the output $\Delta = \sum_{i=1}^{h} \Delta_i + b_0 - b_1$. For the i-th neuron, we denote the high-probability estimate of Δ_i given in the statement of Theorem A.1 as m_0^i when $\ell(x) = 0$ and m_1^i when $\ell(x) = 1$. By union bound, we have the following corollary.

Corollary 5.2. *Given a vertex* $x \in G$ *with label unrevealed, we have*

$$\mathbb{P}[|\Delta - \mu_j| \leq \delta|\ell(x) = j] \geq 1 - O(\frac{1}{n}), \tag{3}$$

where

$$\mu_j = (\sum_{i=1}^{h} m_j^i) + b_0 - b_1,\ j = 0, 1 \qquad \delta = \sum_{i=1}^{h} |\alpha_i - \alpha_i'||\beta_i - \beta_i'|O(\log^{-\frac{1}{2}} n)\,.$$

For any $\epsilon > 0$, we require the probability of concentration in equation 5 to be at least $1 - \tilde{\epsilon}$, where $\tilde{\epsilon} = o(\epsilon)$. If we set $\tilde{\epsilon} = \epsilon^2$, then we set $1 - O(\frac{1}{n}) \geq 1 - \epsilon^2$, i.e.$n \geq \Omega(\frac{1}{\epsilon})^2$. Our following analysis will be based on this condition.

From Theorem A.1, we have the following result about the value of $m_0^i - m_1^i$.

Corollary 5.3.

- *If the* i*-the neuron is of "good type" and satisfies (*G_1*), then*

$$m_0^i - m_1^i = \lambda|\alpha_i - \alpha_i'||\beta_i - \beta_i'|\left(\frac{p-q}{p+q}\right)^2.$$

- *If the* i*-the neuron is of "bad type" and satisfies (*B_1*), then*

$$m_0^i - m_1^i = -\lambda|\alpha_i - \alpha_i'||\beta_i - \beta_i'|\left(\frac{p-q}{p+q}\right)^2.$$

- *If the* i*-the neuron is of "harmless type" and satisfies equation* H_1*, then*

$$m_0^i - m_1^i = \lambda|\beta_i - \beta_i'||p\alpha_i + q\alpha_i'|\frac{p-q}{(p+q)^2}.$$

Similar results for neurons satisfying equation G_2,(B_2), equation B_3 and equation H_1 are stated in the full version of the paper, along with the proof.

Remark.

- As we can see from Corollary 5.3, the value of $m_0^i - m_1^i$ is positive for "good type" neurons, non-negative for "harmless type" neurons and may be negative (but lower bounded) for "bad type" neurons. Since positive values of $m_0^i - m_1^i$ decrease the loss of the model, this explains the names for the types of neurons.

- $m_0^i - m_1^i$ is proportional to $|\alpha_i - \alpha_i'||\beta_i - \beta_i'|$. In the next section, we analyze the dynamics of the parameters $\alpha_i, \alpha_i', \beta_i, \beta_i'$. Using our understanding of these dynamics, in Theorem C.2 we present a refined result about the separation of output which only depends on the initialization of parameters.

- Let $c := \mu_0 - \mu_1 = \sum_{i=1}^{h}(m_0^i - m_1^i)$. By the two corollaries above, we have $\delta = o(|c|)$. The balanced loss guaranteed by the bias term and the coordinate descent scheme ensure that $\mu_0 = \Omega(c)$ and $\mu_1 = \Omega(c)$. It then follows that if the loss is sufficiently small, both μ_0 and μ_1 have correct sign, i.e. $\mu_0 > 0 > \mu_1$. (Otherwise, due to concentration of the output, the model makes wrong prediction and the loss is large). So we will eventually have $\delta = o(\mu_0)$ and $\delta = o(|\mu_1|)$.

6 DYNAMICS OF PARAMETERS

In this section, we describe the dynamics of each type of neurons through coordinate descent, which can be visualized in the following figure in which the arrows indicate movement between types that can happen with non-negligible probability.

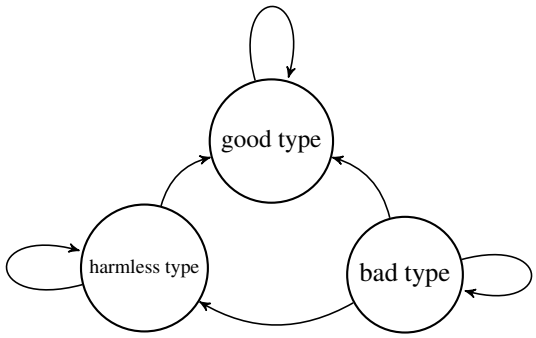

Figure 1: Dynamics of hidden neurons

There are two noteworthy points from this figure. First, "good type" parameters are preserved under coordinate descent. Second, there are no arrows coming into "bad type" except from itself.

These dynamics are proved by estimating the gradient with respect to the loss function for each class of neuron. Because of the non-linearity of the activation, we rely heavily on the concentration result proved above to get tight estimates. Without these concentration results, even estimating the sign of the gradient seems difficult. The proof and experiments about the dynamics of hidden neurons are deferred to the appendix.

7 LEARNING GUARANTEE

In this section, we prove our main result that with high probability a trained GCN can detect communities in SBM with any desired accuracy. The proof is based on the following theorem which shows that if μ_0 and μ_1 are separated enough, then the model achieves high accuracy.

Theorem 7.1. *$\forall \epsilon > 0$, if $\left|\mathbb{E}[\frac{\partial L}{\partial b_0}]\right| < \frac{\epsilon}{4}$ and $\sigma(-\frac{\mu_0-\mu_1}{2}) < \frac{\epsilon}{2}$, then $\mathbb{P}[\Delta < 0|\ell(x) = 0] < 4\epsilon, \mathbb{P}[\Delta > 0|\ell(x) = 1] < 4\epsilon$, where $\sigma(x) := \frac{1}{1+\exp(-x)}$ represents the sigmoid function.*

Next we show that the model can achieve such separation between μ_0 and μ_1 through coordinate descent. In order to make constant update of parameters at every epoch, we set an adaptive learning rate $\eta_k = \frac{1}{\mathbb{E}[Z^{(k)}]}$. We first refine Corollary 5.3 about the separation of output for each type of neuron $(m_0^i - m_1^i)$ using the dynamics of parameters.

Theorem 7.2 (separation of output)**.** *Let m_0^i and m_1^i be defined as in Section 5, train the model for k epochs by the defined coordinate descent with adaptive learning rate $\eta_k = \frac{1}{\mathbb{E}[Z^{(k)}]}$,*

- *if the i-th neuron is of "good type", then*

$$m_0^i - m_1^i \geq A_i^{(0)} B_i^{(0)} \frac{\lambda}{2} \left(\frac{p-q}{p+q}\right)^2 \left(1 + \frac{\sqrt{2}\lambda}{8}\left(\frac{p-q}{p+q}\right)^2\right)^{2k}$$

- *if the i-th neuron is of "bad type", then*

$$m_0^i - m_1^i \geq -k\big(\big(A_i^{(0)}\big)^2 + \big(B_i^{(0)}\big)^2\big)\frac{\lambda p(p-q)}{(p+q)^2},$$

- *if the i-th neuron is of "harmless type", then*

$$m_0^i - m_1^i \geq 0,$$

where $A_i^{(0)} = \alpha_i^{(0)} - \alpha_i'^{(0)}, B_i^{(0)} = \beta_i^{(0)} - \beta_i'^{(0)}$.

Next we present a result about initialization, which shows that with high probability, there are enough "good type" neurons and parameters have appropriate scale.

Lemma 7.3. *Suppose all parameters in* $W^{(0)}$ *and* $W^{(1)}$ *are initialized independently following standard normal distribution. Then the number* h_g *of neurons initialized as "good type" satisfies* $\mathbb{P}[h_g \geq \frac{h}{8}] \geq 1 - \exp(-\frac{h}{64})$. *Furthermore,*

$$\mathbb{P}[\sum_{i=1}^{h}(\alpha_i-\alpha_i')^2+(\beta_i-\beta_i')^2 \leq 5h] \geq 1-O(\frac{1}{h}), \qquad \mathbb{P}[\sum_{\substack{\text{the } i\text{-th neuron} \\ \text{initialized as} \\ \text{"good type"}}} |\alpha_i-\alpha_i'||\beta_i-\beta_i'| \geq \frac{h}{80}] \geq 1-O(\frac{1}{h}).$$

Now we can prove the final result.

Proof of Theorem 2.2. First we show that if the loss $\mathbb{E}[Z]$ is small enough, the model achieves desired accuracy. Indeed, if $\mathbb{E}[Z] < 2\epsilon$, since

$$\mathbb{E}[Z] = \mathbb{E}[Z|\text{pred is wrong}]\mathbb{P}[\text{pred is wrong}]+\mathbb{E}[Z|\text{pred is correct}]\mathbb{P}[\text{pred is correct}] \geq \frac{1}{2}\mathbb{P}[\text{pred is wrong}],$$

we have $\mathbb{P}[\text{pred is wrong}] \leq 4\epsilon$, i.e., $\mathbb{P}[\text{pred is correct}] > 1 - 4\epsilon$.

Otherwise, $\mathbb{E}[Z] \geq 2\epsilon$, since $\mathbb{E}[Z] = \frac{1}{2}(\mathbb{E}[Z|\ell(z) = 0] + \mathbb{E}[Z|\ell(z) = 1])$, we have $\mathbb{E}[Z|\ell(z) = 0]+\mathbb{E}[Z|\ell(z) = 1] \geq 4\epsilon$. On the other hand, $\left|\mathbb{E}[\frac{\partial L}{\partial b_0}]\right| < \epsilon$ implies that $\left|\mathbb{E}[Z|\ell(z) = 0]-\mathbb{E}[Z|\ell(z) = 1]\right| < 2\epsilon$.

By Theorem C.2,

$$\begin{aligned}\mu_0 - \mu_1 = \sum_{i=1}^{h}(m_0^i - m_1^i) &= \sum_{i\in\text{"good"}}(m_0^i - m_1^i) + \sum_{i\in\text{"bad"}}(m_0^i - m_1^i) + \sum_{i\in\text{"harmless"}}(m_0^i - m_1^i) \\ &\geq \frac{\lambda}{2}\left(\frac{p-q}{p+q}\right)^2\left(1+\frac{\sqrt{2}\lambda}{8}\left(\frac{p-q}{p+q}\right)^2\right)^{2k}\sum_{i\in\text{"}good''}A_i^{(0)}B_i^{(0)} - k\frac{\lambda p(p-q)}{(p+q)^2}\sum_{i\in\text{"}bad''}\left(\left(A_i^{(0)}\right)^2+\left(B_i^{(0)}\right)^2\right).\end{aligned}$$

$$\sum_{i\in\text{"good"}}A_i^{(0)}B_i^{(0)} \geq \frac{h}{80}, \qquad \sum_{i\in\text{"bad"}}\left(\left(A_i^{(0)}\right)^2+\left(B_i^{(0)}\right)^2\right) \leq 5h.$$

Then with probability $\geq 1 - O(\frac{1}{h})$,

$$\begin{aligned}\mu_0 - \mu_1 &\geq h\left(\frac{\lambda}{160}\left(\frac{p-q}{p+q}\right)^2\left(1+\frac{\sqrt{2}\lambda}{8}\left(\frac{p-q}{p+q}\right)^2\right)^{2k} - k\frac{5\lambda p(p-q)}{(p+q)^2}\right) \\ &\geq h(C_1(1+C_2)^{2k} - C_3k), \end{aligned} \tag{4}$$

where C_1, C_2 and C_3 are constants determined by p, q and λ.

By Theorem C.1, if (4)$\geq 2\log\frac{2}{\epsilon}$ (then $\sigma(-\frac{\mu_0-\mu_1}{2}) \leq \frac{\epsilon}{2}$), then the model achieves accuracy $\geq 1 - 4\epsilon$. It's sufficient to have

$$C_1(1+C_2)^{2k} - C_3k \geq 2\log\frac{2}{\epsilon},$$

i.e. $k = O(\log\log\frac{1}{\epsilon})$. □

8 EXPERIMENTS

We show some experiments verifying Theorem 2.2. In particular, our experiments demonstrate that accuracy increases with n, the probability of high-accuracy models increases with h, and coordinate descent is able to recovery high-accuracy models in the sparse regime of Assumption 2.1. Additional plots demonstrating the dynamics of hidden neurons with their ratios and differences can be seen in the appendix.

Experiment 1 In this experiment, we plot the an estimate of the accuracy versus epoch for varying n. The parameters p, q of SBM follow Assumption 2.1, where we choose $a = 1.01$ and $b = 0.67$ for $n = 250$, $a = 1.29$ and $b = 0.86$ for $n = 500$, $a = 1.70$ and $b = 1.13$ for $n = 1000$. We set $h = 20, \lambda = 0.3$. For each n, we run 40 independent experiments. In each experiment we train the model for 100 epochs. The training set has 40 randomly generated graphs from SBM(n, p, q). We validate the performance by the percentage of correct predictions on 200 random vertices, each from a randomly generated graph. The result is shown Figure 2. The shaded region for each n is obtained from the max, min and mean percentage of the 40 experiments. As shown, the percentage increases uniformly with n.

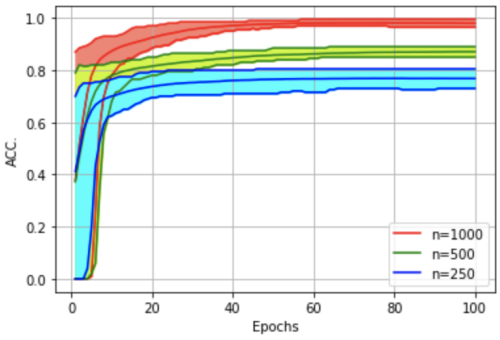

Figure 2: Display of accuracy versus epoch for varying n. Accuracy is computed on vertices drawn from random SBM graphs. The accuracy increases with epoch, and we also see that the lower bound on the accuracy increases with n, as expected from our theory.

Experiment 2 In this experiment, we show the effect of the number of hidden neurons h. The parameters of SBM are the same as Experiment 1. We set $h = 2, 5, 20$. For each pair of (n, h) we run 40 independent experiments and show the distribution of validation in Figure 3. In each plot, the x-axis represents the accuracy, while y-axis represents the count of experiments. We can see that in each row of Figure 3, as h increases, we have lager probability to achieve high accuracy; in each column, as n increases, the model achieves higher accuracy. The results verify our theory in the paper.

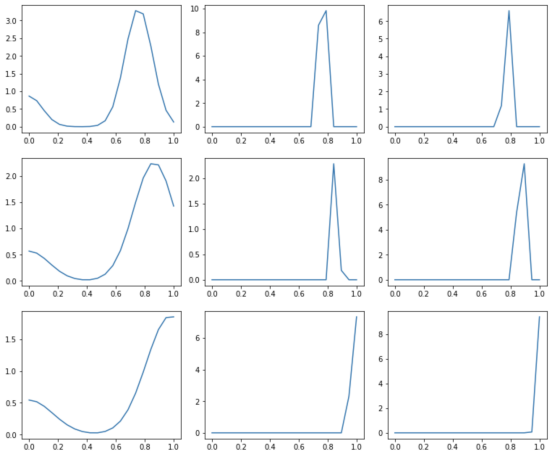

Figure 3: Display of accuracy for varying n and h. For each row, n is set to be 250 (top), 500 (middle), 1000 (bottom); for each column, h is set to be 2 (left), 5 (middle), 20 (right). Probability of failure decreases as h increases, and accuracy increases with n.

9 Future Directions

Graph neural networks offer a promising setting for progress on the more general theory of deep learning, because random graph models more plausibly capture the structure of real-world data compared to, e.g., the Gaussian inputs often used to prove deep learning guarantees for traditional feed-forward neural networks. This paper has initiated the project of proving training guarantees for semi-supervised learning using GCNs on SBM models, but much more work remains to be done. Arguably the sparsest SBM models (expected constant degree) are the most compelling from the perspective of modeling real-world communities, so it would be interesting to extend these results to that setting. Models with more than two blocks, or overlapping communities (Petti & Vempala, 2018) would be even closer to real-world structure. We hope this initial step spurs further interest in provable guarantees for training neural networks using plausible models of real-world data as the input distribution.

REFERENCES

Emmanuel Abbe and Colin Sandon. Recovering communities in the general stochastic block model without knowing the parameters. *arXiv preprint arXiv:1506.03729*, 2015.

Alon Brutzkus and Amir Globerson. Globally optimal gradient descent for a convnet with gaussian inputs. In *International conference on machine learning*, pp. 605–614. PMLR, 2017.

Jianfei Chen, Jun Zhu, and Le Song. Stochastic training of graph convolutional networks with variance reduction. *arXiv preprint arXiv:1710.10568*, 2017.

Jie Chen, Tengfei Ma, and Cao Xiao. Fastgcn: fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247*, 2018.

Sitan Chen, Adam R. Klivans, and Raghu Meka. Learning deep relu networks is fixed-parameter tractable, 2020a.

Zhengdao Chen, Xiang Li, and Joan Bruna. Supervised community detection with line graph neural networks, 2020b.

Mikhail Drobyshevskiy and Denis Turdakov. Random graph modeling: A survey of the concepts. *ACM Comput. Surv.*, 52(6):1–36, December 2019.

Hongyang Gao, Zhengyang Wang, and Shuiwang Ji. Large-scale learnable graph convolutional networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1416–1424, 2018.

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

Rong Ge, Jason D. Lee, and Tengyu Ma. Learning one-hidden-layer neural networks with landscape design. *CoRR*, abs/1711.00501, 2017.

Surbhi Goel, Aravind Gollakota, Zhihan Jin, Sushrut Karmalkar, and Adam Klivans. Superpolynomial lower bounds for learning one-layer neural networks using gradient descent. In *International Conference on Machine Learning*, pp. 3587–3596. PMLR, 2020.

Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

Tatsuro Kawamoto, Masashi Tsubaki, and Tomoyuki Obuchi. Mean-field theory of graph neural networks in graph partitioning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12): 124007, dec 2019. doi: 10.1088/1742-5468/ab3456. URL `https://doi.org/10.1088/1742-5468/ab3456`.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.

R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Stochastic models for the web graph. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pp. 57–65, 2000. doi: 10.1109/SFCS.2000.892065.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4602–4609, 2019.

Samantha Petti and Santosh S. Vempala. Approximating sparse graphs: The random overlapping communities model, 2018.

Ryoma Sato. A survey on the expressive power of graph neural networks, 2020.

Ryoma Sato, Makoto Yamada, and Hisashi Kashima. Approximation ratios of graph neural networks for combinatorial problems. *arXiv preprint arXiv:1905.10261*, 2019.

Santosh Vempala and John Wilmes. Gradient descent for one-hidden-layer neural networks: Polynomial convergence and sq lower bounds. In *Conference on Learning Theory*, pp. 3115–3117. PMLR, 2019.

Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021.

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

Prateek Yadav, Madhav Nimishakavi, Naganand Yadati, Shikhar Vashishth, Arun Rajkumar, and Partha Talukdar. Lovasz convolutional networks. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1978–1987. PMLR, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Xiao Zhang, Yaodong Yu, Lingxiao Wang, and Quanquan Gu. Learning one-hidden-layer relu networks via gradient descent. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1524–1534. PMLR, 2019a.

Yingxue Zhang, Soumyasundar Pal, Mark Coates, and Deniz Ustebay. Bayesian graph convolutional neural networks for semi-supervised classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):5829–5836, 2019b.

Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*, 2018.

A Concentration and Separation of Output

Let $N(x,0), N(x,1)$ denote the neighborhood of vertex x with label 0 and 1 respectively, i.e. $N(x,0) = \{y \in G, y \sim x, \ell(y) = 0\}, N(x,1) = \{y \in G, y \sim x, \ell(y) = 1\}$. By the definition of SBM, both $|N(x,0)|$ and $|N(x,1)|$ are binomial random variables. For $\ell(x) = 0, |N(x,0)| \sim B(\frac{n}{2}, p), |N(x,1)| \sim B(\frac{n}{2}, q)$ and for $\ell(x) = 1, |N(x,0)| \sim B(\frac{n}{2}, q), |N(x,1)| \sim B(\frac{n}{2}, p)$. Moreover, t_0^y and t_1^y are also binomial random variables, for $\ell(y) = 0, t_0^y \sim B(\frac{n\lambda}{2}, p), t_1^y \sim B(\frac{n\lambda}{2}, q)$, similarly for $\ell(y) = 1$. Our following analysis is based on the condition that $|N(x,0)|, |N(x,1)|, t_0^x$ and t_1^x are in their high probability range for all $x \in G$. Specifically we require the condition that for all $\ell(x) = 0$, (similar conditions for $\ell(x) = 1$ are omitted):

$$\left||N(x,0)| - \frac{np}{2}\right| \leq O(np)^{\frac{5}{6}}, \left||N(x,1)| - \frac{nq}{2}\right| \leq O(nq)^{\frac{5}{6}}; \tag{Cond}$$
$$\left|t_0^x - \frac{n\lambda p}{2}\right| \leq O(np)^{\frac{5}{6}}, \left|t_1^x - \frac{n\lambda q}{2}\right| \leq O(nq)^{\frac{5}{6}}.$$

By tail bound of binomial random variables and union bound, we have

$$\mathbb{P}[(Cond)] \geq 1 - \frac{1}{n^2}$$

Under this condition, we show the concentration of Δ_i for each type.

A.1 "Good Type" neurons

For convenience, according to the activation pattern of $\phi(\alpha_i t_0^y + \alpha_i' t_1^y)$, we further divide (G_2) into subcases $(G_{2,1})$, $(G_{2,2})$ and $(G_{2,3})$ by to the ratio of $\left|\frac{\alpha_i}{\alpha_i'}\right|$. For example, in (G_1) and $(G_{2,1})$, $\phi(\alpha_i t_0^y + \alpha_i' t_1^y)$ is active for both $\ell(y) = 0$ and $\ell(y) = 1$; in $(G_{2,1})$, it is only active for $\ell(y) = 0$.

$$\left|\frac{\alpha_i}{\alpha_i'}\right| > \frac{p}{q}(1+\log^{-\frac{1}{3}} n) \tag{$G_{2,1}$}$$

$$1 < \left|\frac{\alpha_i}{\alpha_i'}\right| < \frac{p}{q}(1-\log^{-\frac{1}{3}} n) \tag{$G_{2,2}$}$$

$$\frac{p}{q}(1-\log^{-\frac{1}{3}} n) \leq \left|\frac{\alpha_i}{\alpha_i'}\right| \leq \frac{p}{q}(1+\log^{-\frac{1}{3}} n) \tag{$G_{2,3}$}$$

We have the following estimation of Δ_i in "good type".

Theorem A.1 (concentration of output from "good type" neurons)**.** *If the i-th neuron is of "good type", then*

- *in both (G_1) and ($G_{2,1}$):*

$$\mathbb{P}[\big|\Delta_i - \frac{\lambda(\beta_i-\beta_i')}{(p+q)^2}[(p^2+q^2)\alpha_i + 2pq\alpha_i']\big| \leq (\alpha_i-\alpha_i')(\beta_i-\beta_i')O(\log^{-\frac{1}{2}} n)|\ell(x)=0] \geq 1-O(\frac{1}{n^2})$$

$$\mathbb{P}[\big|\Delta_i - \frac{\lambda(\beta_i-\beta_i')}{(p+q)^2}[2pq\alpha_i + (p^2+q^2)\alpha_i']\big| \leq (\alpha_i-\alpha_i')(\beta_i-\beta_i')O(\log^{-\frac{1}{2}} n)|\ell(x)=1] \geq 1-O(\frac{1}{n^2})$$

- *in ($G_{2,2}$):*

$$\mathbb{P}[\big|\Delta_i - \frac{\lambda(\beta_i-\beta_i')p}{(p+q)^2}(p\alpha_i + q\alpha_i')\big| \leq (\alpha_i-\alpha_i')(\beta_i-\beta_i')O(\log^{-\frac{1}{2}} n)|\ell(x)=0] \geq 1-O(\frac{1}{n^2})$$

$$\mathbb{P}[\big|\Delta_i - \frac{\lambda(\beta_i-\beta_i')q}{(p+q)^2}(p\alpha_i + q\alpha_i')\big| \leq (\alpha_i-\alpha_i')(\beta_i-\beta_i')O(\log^{-\frac{1}{2}} n)|\ell(x)=1] \geq 1-O(\frac{1}{n^2})$$

- *in ($G_{2,3}$):*

$$\mathbb{P}[\big|\Delta_i - \frac{\lambda(\beta_i-\beta_i')p}{(p+q)^2}(p\alpha_i + q\alpha_i')\big| \leq (\alpha_i-\alpha_i')(\beta_i-\beta_i')O(\log^{-\frac{1}{2}} n)|\ell(x)=0] \geq 1-O(\frac{1}{n^2})$$

$$\mathbb{P}[\big|\Delta_i - \frac{\lambda(\beta_i-\beta_i')q}{(p+q)^2}(p\alpha_i + q\alpha_i')\big| \leq (\alpha_i-\alpha_i')(\beta_i-\beta_i')O(\log^{-\frac{1}{2}} n)|\ell(x)=1] \geq 1-O(\frac{1}{n^2}).$$

Proof. We have $\Delta_i = (\beta_i - \beta_i')\sum_{y\in G}\mathbb{1}[y\sim x]\frac{4\phi(\alpha_i t_0^y + \alpha_i' t_1^y)}{n^2(p+q)^2}$. We apply the method of averaged bounded difference [book] to estimate Δ_i. In different parameter regimes, $\phi(\alpha_i t_0^y + \alpha_i' t_1^y)$ has different activation patterns.

In (G_1) and $(G_{2,1})$, $\phi(\alpha_i t_0^y + \alpha_i' t_1^y)$ is active with probability $1 - O(\frac{1}{n^2})$ for both $\ell(y) = 0$ and $\ell(y) = 1$. For $\ell(x) = 0$, at first we estimate $\mathbb{E}[\Delta_i]$. By condition (Cond):

$$\left|\mathbb{E}[\Delta_i] - \frac{\lambda(\beta_i - \beta_i')}{(p+q)^2}[(p^2+q^2)\alpha_i + 2pq\alpha_i']\right| \leq (\alpha_i - \alpha_i')(\beta_i - \beta_i')O(\log^{-1} n).$$

Let $Y_j = \frac{4\phi(\alpha_i t_0^{y_j} + \alpha_i' t_1^{y_j})}{n^2(p+q)^2}$, then $\Delta_i = (\beta_i - \beta_i')\sum_j Y_j$. Based on condition (Cond, $\left|Y_j - \frac{2\lambda(p\alpha_i + q\alpha_i')}{n(p+q)^2}\right| \leq (\alpha_i - \alpha_i')O(\log^{-\frac{7}{2}} n)$ for $\ell(y_j) = 0$. For any a_k, a_k',

$$\left|\mathbb{E}[Y_k|Y_1, \cdots, Y_{k-1}, Y_k = a_k] - \mathbb{E}[Y_k|Y_1, \cdots, Y_{k-1}, Y_k = a_k']\right| \leq (\alpha_i - \alpha_i')O(\log^{-\frac{7}{2}} n).$$

Moreover, when the number of vertices with revealed labels are fixed,

$$\left|\mathbb{E}[Y_j|Y_1, \cdots, Y_{k-1}, Y_k = a_k] - \mathbb{E}[Y_j|Y_1, \cdots, Y_{k-1}, Y_k = a_k']\right| \leq (\alpha_i - \alpha_i')O(\log^{-6} n),$$

for $j \geq k$. By condition equation Cond, there are at most $O(\log^3 n)$ non-zero terms for $Y_k, 1 \leq k \leq n$. So

$$\left|\mathbb{E}[\sum_j Y_j|Y_1, \cdots, Y_{k-1}, Y_k = a_k] - \mathbb{E}[\sum_j Y_j|Y_1, \cdots, Y_{k-1}, Y_k = a_k']\right| \leq (\alpha_i - \alpha_i')O(\log^{-\frac{7}{2}} n),$$

for $1 \leq k \leq n$. By the method of averaged bounded difference, we have

$$\mathbb{P}[\left|\Delta_i - \frac{\lambda(\beta_i - \beta_i')}{(p+q)^2}[(p^2+q^2)\alpha_i + 2pq\alpha_i']\right| \leq (\alpha_i - \alpha_i')(\beta_i - \beta_i')O(\log n)^{-\frac{1}{2}}|\ell(x) = 0] \geq 1 - O(\frac{1}{n^2}).$$

Other regimes can be proved similarly. □

A.2 "Bad Type" Neurons

For convenience of our analysis, we further divide B_2 into subcases $B_{2,1}$, $B_{2,2}$and $B_{2,3}$ according to the ratio of $\left|\frac{\alpha_i}{\alpha_i'}\right|$

$$\left|\frac{\alpha_i}{\alpha_i'}\right| > \frac{p}{q}(1 + \log^{-\frac{1}{3}} n) \tag{$B_{2,1}$}$$

$$\left|\frac{\alpha_i}{\alpha_i'}\right| \in (\frac{q}{p}(1 + \log^{-\frac{1}{3}} n), \frac{p}{q}(1 - \log^{-\frac{1}{3}} n)] \tag{$B_{2,2}$}$$

$$\left|\frac{\alpha_i}{\alpha_i'}\right| \in (\frac{p}{q}(1 - \log^{-\frac{1}{3}} n), \frac{p}{q}(1 + \log^{-\frac{1}{3}} n)] \tag{$B_{2,3}$}$$

We have the following estimation of Δ_i in "bad type". Without loss of generality, we only consider the cases outside the parenthesis.

Theorem A.2 (concentration of output from "bad type" neurons)**.** *If the i-th neuron is of "bad type", we have:*

- *in B_1 and $B_{2,1}$:*

$$\mathbb{P}[\left|\Delta_i - \frac{\lambda(\beta_i - \beta_i')}{(p+q)^2}[(p^2+q^2)\alpha_i + 2pq\alpha_i']\right| \leq |\alpha_i - \alpha_i'||\beta_i - \beta_i'|O(\log^{-\frac{1}{2}} n)|\ell(x) = 0] \geq 1 - O(\frac{1}{n^2})$$

$$\mathbb{P}[\left|\Delta_i - \frac{\lambda(\beta_i - \beta_i')}{(p+q)^2}[2pq\alpha_i + (p^2+q^2)\alpha_i']\right| \leq |\alpha_i - \alpha_i'||\beta_i - \beta_i'|O(\log^{-\frac{1}{2}} n)|\ell(x) = 1] \geq 1 - O(\frac{1}{n^2})$$

- *in* $B_{2,2}$*:*

$$\mathbb{P}[\big|\Delta_i - \frac{\lambda(\beta_i - \beta_i')p}{(p+q)^2}(p\alpha_i + q\alpha_i')\big| \leq |\alpha_i - \alpha_i'||\beta_i - \beta_i'|O(\log^{-\frac{1}{2}} n)|\ell(x) = 0] \geq 1 - O(\frac{1}{n^2})$$

$$\mathbb{P}[\big|\Delta_i - \frac{\lambda(\beta_i - \beta_i')q}{(p+q)^2}(p\alpha_i + q\alpha_i')\big| \leq |\alpha_i - \alpha_i'||\beta_i - \beta_i'|O(\log^{-\frac{1}{2}} n)|\ell(x) = 1] \geq 1 - O(\frac{1}{n^2})$$

- *in* $B_{2,3}$*:*

$$\mathbb{P}[\big|\Delta_i - \frac{\lambda(\beta_i - \beta_i')p}{(p+q)^2}(p\alpha_i + q\alpha_i')\big| \leq |\alpha_i - \alpha_i'||\beta_i - \beta_i'|O(\log^{-\frac{1}{2}} n)|\ell(x) = 0] \geq 1 - O(\frac{1}{n^2})$$

$$\mathbb{P}[\big|\Delta_i - \frac{\lambda(\beta_i - \beta_i')q}{(p+q)^2}(p\alpha_i + q\alpha_i')\big| \leq |\alpha_i - \alpha_i'||\beta_i - \beta_i'|O(\log^{-\frac{1}{2}} n)|\ell(x) = 1] \geq 1 - O(\frac{1}{n^2}).$$

- *in* B_3*:*

$$\mathbb{P}[\big|\Delta_i\big| \leq |\alpha_i - \alpha_i'||\beta_i - \beta_i'|O(\log^{-\frac{1}{2}} n)|\ell(x) = 0 \textit{ or } 1] \geq 1 - O(\frac{1}{n^2}).$$

Proof. The proof is similar as Theorem A.1 □

A.3 "HARMLESS TYPE" NEURONS

We have the following estimation of Δ_i in "harmless type".

Theorem A.3 (concentration of output from "harmless type" neurons)**.** *If the i-th neuron is of "harmless type", we have:*

- *in* H_1*:*

$$\mathbb{P}[\big|\Delta_i - \frac{\lambda(\beta_i - \beta_i')p}{(p+q)^2}(p\alpha_i + q\alpha_i')\big| \leq (\alpha_i - \alpha_i')(\beta_i - \beta_i')O(\log^{-\frac{1}{2}} n)|\ell(x) = 0] \geq 1 - O(\frac{1}{n^2})$$

$$\mathbb{P}[\big|\Delta_i - \frac{\lambda(\beta_i - \beta_i')q}{(p+q)^2}(p\alpha_i + q\alpha_i')\big| \leq (\alpha_i - \alpha_i')(\beta_i - \beta_i')O(\log^{-\frac{1}{2}} n)|\ell(x) = 1] \geq 1 - O(\frac{1}{n^2})$$

- *in* H_2*:* $\Delta_i = 0$ *for both* $\ell(x) = 0$ *and 1.*

A.4 CONCENTRATION AND SEPARATION OF OUTPUT

Corollary A.4. *Given a vertex $x \in G$ with label unrevealed, we have*

$$\mathbb{P}[|\Delta - \mu_j| \leq \delta|\ell(x) = j] \geq 1 - \exp(-\Omega(n^{\frac{1}{3}})), \tag{5}$$

where $\mu_j = (\sum_{i=1}^h m_j^i) + b_0 - b_1, j = 0, 1, \delta = \sum_{i=1}^h |\alpha_i - \alpha_i'||\beta_i - \beta_i'|O(n^{-\frac{1}{6}+\tau})$.

From Theorem A.1, A.2 and A.3, we have the following result about the value of $m_0^i - m_1^i$ by straightforward computation.

B DYNAMICS OF PARAMETERS

We consider the cross-entropy loss in training. The loss on a particular vertex x is

$$L(x) = -\log O_{\ell(x)}(x),$$

where $O_0(x)$ and $O_1(x)$ are the first and second component of the output respectively, i.e.

$$O_0(x) = \frac{\exp(g_0(x))}{\exp(g_0(x)) + \exp(g_1(x))}, \quad O_1(x) = \frac{\exp(g_1(x))}{\exp(g_0(x)) + \exp(g_1(x))}.$$

For a given graph G generated by SBM, we set the objective function $L(G)$ as the average loss over all the vertices with revealed labels[1], i.e.

$$L(G) = \frac{1}{\#\{x \in G : \ell(x) \text{ is revealed}\}} \sum_{x:\ell(x) \text{ revealed}} L(x).$$

We first show the partial derivatives of parameters.

Theorem B.1 (derivatives of parameters)**.** *For $1 \le i \le h$, let x be a vertex, $\ell(x)$ its true label, $L(x) = -\log O_{\ell(x)}(x)$, then*

$$\begin{aligned}
\frac{\partial L}{\partial \alpha_i} &= \frac{4}{n^2(p+q)^2}(\beta_i - \beta_i')Z(-1)^{1-\ell(x)} \sum_y \mathbb{1}[y \sim x]\mathbb{1}[\alpha_i t_0^y + \alpha_i' t_1^y \ge 0] t_0^y \\
\frac{\partial L}{\partial \alpha_i'} &= \frac{4}{n^2(p+q)^2}(\beta_i - \beta_i')Z(-1)^{1-\ell(x)} \sum_y \mathbb{1}[y \sim x]\mathbb{1}[\alpha_i t_0^y + \alpha_i' t_1^y \ge 0] t_1^y \\
\frac{\partial L}{\partial \beta_i} &= \frac{4}{n^2(p+q)^2} Z(-1)^{1-\ell(x)} \sum_y \mathbb{1}[y \sim x]\phi(\alpha_i t_0^y + \alpha_i' t_1^y) \\
\frac{\partial L}{\partial \beta_i'} &= -\frac{4}{n^2(p+q)^2} Z(-1)^{1-\ell(x)} \sum_y \mathbb{1}[y \sim x]\phi(\alpha_i t_0^y + \alpha_i' t_1^y) \\
\frac{\partial L}{\partial b_0} &= (-1)^{1-\ell(x)} Z \\
\frac{\partial L}{\partial b_1} &= (-1)^{\ell(x)} Z,
\end{aligned}$$

where $Z = \frac{\exp(g_{1-\ell(x)}(x))}{\exp(g_0(x))+\exp(g_1(x))}$, t_0^y and t_1^y are the numbers of neighbors of y(including perhaps y itself) with revealed labels in class 0 and class 1 respectively.

Proof. We compute $\frac{\partial L}{\partial \alpha_i}$, $\frac{\partial L}{\partial \beta_i}$ and $\frac{\partial L}{\partial b_0}$, others can be computed symmetrically. We have

$$L(x) = -\log O_{\ell(x)}(x) = \log(\exp(g_0(x)) + \exp(g_1(x))) - g_{\ell(x)}(x),$$

since $O_j(x) = \frac{\exp(g_j(x))}{\exp(g_0(x))+\exp(g_1(x))}$, $j = 0, 1$. So

$$\begin{aligned}
\frac{\partial L}{\partial \alpha_i} &= \frac{e^{g_0(x)}\frac{\partial g_0(x)}{\partial \alpha_i} + e^{g_1(x)}\frac{\partial g_1(x)}{\partial \alpha_i}}{e^{g_0(x)} + e^{g_1(x)}} - \frac{\partial g_{\ell(x)}(x)}{\partial \alpha_i} \\
&= (-1)^{1-\ell(x)} Z\big(\frac{\partial g_0(x)}{\partial \alpha_i} - \frac{\partial g_1(x)}{\partial \alpha_i}\big).
\end{aligned}$$

Since $g_j(x) = f_j(x) + b_j$, $\frac{\partial g_j(x)}{\partial \alpha_i} = \frac{\partial f_j(x)}{\partial \alpha_i}$, $j = 0, 1$. By (2)

$$\begin{aligned}
\frac{\partial f_0(x)}{\partial \alpha_i} &= \frac{4}{n^2(p+q)^2} \sum_y \mathbb{1}[y \sim x]\beta_i \mathbb{1}[\alpha_i t_0^y + \alpha_i' t_1^y \ge 0] t_0^y \\
\frac{\partial f_1(x)}{\partial \alpha_i} &= \frac{4}{n^2(p+q)^2} \sum_y \mathbb{1}[y \sim x]\beta_i' \mathbb{1}[\alpha_i t_0^y + \alpha_i' t_1^y \ge 0] t_0^y.
\end{aligned}$$

[1] We abuse the notation L for $L(x)$ and $L(G)$, but the meaning is clear from the context.

Therefore

$$\frac{\partial L}{\partial \alpha_i} = \frac{4}{n^2(p+q)^2}(-1)^{1-\ell(x)} Z(\beta_i - \beta_i') \sum_y \mathbb{1}[y \sim x]\mathbb{1}[\alpha_i t_0^y + \alpha_i' t_1^y \geq 0] t_0^y.$$

Next we compute $\frac{\partial L}{\partial \beta_i}$. Similar as above, $\frac{\partial L}{\partial \beta_i} = (-1)^{1-\ell(x)} Z\big(\frac{\partial f_0(x)}{\partial \beta_i} - \frac{\partial f_1(x)}{\partial \beta_i}\big)$. By (2)

$$\frac{\partial f_0(x)}{\partial \beta_i} = \frac{4}{n^2(p+q)^2} \sum_y \mathbb{1}[y \sim x]\phi(\alpha_i t_0^y + \alpha_i' t_1^y)$$

$$\frac{\partial f_1(x)}{\partial \alpha_i} = 0.$$

So

$$\frac{\partial L}{\partial \beta_i} = \frac{4}{n^2(p+q)^2}(-1)^{1-\ell(x)} Z \sum_y \mathbb{1}[y \sim x]\phi(\alpha_i t_0^y + \alpha_i' t_1^y).$$

Lastly,

$$\begin{aligned}\frac{\partial L}{\partial b_0} &= (-1)^{1-\ell(x)} Z\left(\frac{\partial g_0(x)}{\partial b_0} - \frac{\partial g_1(x)}{\partial b_0}\right) \\ &= (-1)^{1-\ell(x)} Z,\end{aligned}$$

since $\frac{\partial g_0(x)}{\partial b_0} = 1$, $\frac{\partial g_1(x)}{\partial b_0} = 0$. □

In the following, we will use Theorem B.1 to analyze the dynamics of neurons of each type. As we can see, all of $\frac{\partial L}{\partial \alpha_i}$, $\frac{\partial L}{\partial \alpha_i'}$, $\frac{\partial L}{\partial \beta_i}$ and $\frac{\partial L}{\partial \beta_i'}$ have the form YZ. In order to estimate these derivatives, we show the concentration of Y and Z respectively. To estimate the concentration of Z, we need the concentration of output obtained in Section 5. For any $\epsilon > 0$, we require the probability of concentration in (5) to be at least $1 - \tilde{\epsilon}$, where $\tilde{\epsilon} = o(\epsilon)$. In particular, if we set $\tilde{\epsilon} = \epsilon^2$, then we set $1 - O(\frac{1}{n}) \geq 1 - \epsilon^2$, i.e.

$$n \geq \Omega\left(\frac{1}{\epsilon}\right)^2. \tag{6}$$

Our following analysis will be based on this condition.

Meanwhile in order to have balanced performance in each epoch of coordinate descent, we require $\big|\mathbb{E}[\frac{\partial L}{\partial b_0}]\big| < \frac{\tilde{\epsilon}}{2}$. Since $\mathbb{E}[\frac{\partial L}{\partial b_0}] = \frac{1}{2}(-\mathbb{E}[Z|\ell(x)=0] + \mathbb{E}[Z|\ell(x)=1]))$, we have

$$\big|\mathbb{E}[Z|\ell(x)=0] - \mathbb{E}[Z|\ell(x)=1]\big| < \tilde{\epsilon}. \tag{7}$$

We have the following relation between μ_0 and μ_1. In the following, σ represents the sigmoid function: $\sigma(x) = \frac{1}{1+e^{-x}}$.

Proposition B.2. *If $\big|\mathbb{E}[Z|\ell(x)=0] - \mathbb{E}[Z|\ell(x)=1]\big| < \tilde{\epsilon}$, then $|\sigma(-\mu_0) - \sigma(\mu_1)| \leq \sigma'(\mu_0 - \delta)\delta + \sigma'(\mu_1 + \delta)\delta + 3\tilde{\epsilon}$, where δ and $\tilde{\epsilon}$ are as stated above.*

Proof. We have

$$Z = \begin{cases} \sigma(-\Delta), & \ell(x) = 0 \\ \sigma(\Delta), & \ell(x) = 1 \end{cases}$$

For $\ell(x) = 0$, by Lagrange mean value theorem, $|\sigma(-\mu_0) - Z| = |\sigma(-\mu_0) - \sigma(-\Delta)| = \sigma'(\xi)(\Delta - \mu_0)$, where ξ is between $-\mu_0$ and $-\Delta$. By Corollary A.4 and the condition of n, $|\Delta - \mu_0| \leq \delta$ with probability $\geq 1 - \tilde{\epsilon}$. From the remark following Corollary 5.3, we have $\sigma'(\xi) \leq \sigma'(-\mu_0 + \delta) = \sigma'(\mu_0 - \delta)$,[2] with probability $\geq 1 - \tilde{\epsilon}$. Then we have

$$\mathbb{P}[|\sigma(-\mu_0) - Z| \leq \sigma'(\mu_0 - \delta)\delta | \ell(x) = 0] \geq 1 - \tilde{\epsilon}.$$

[2] $\sigma'(x)$ is even.

Since

$$\mathbb{E}[Z|\ell(x)=0] = \mathbb{E}[Z||\sigma(-\mu_0)-Z| \le \sigma'(\mu_0-\delta)\delta]\mathbb{P}[|\sigma(-\mu_0)-Z| \le \sigma'(\mu_0-\delta)\delta] \\ + \mathbb{E}[Z||\sigma(-\mu_0)-Z| > \sigma'(\mu_0-\delta)\delta]\mathbb{P}[|\sigma(-\mu_0)-Z| > \sigma'(\mu_0-\delta)\delta],$$

then (note that $0 < Z < 1$)

$$\mathbb{E}[Z|\ell(x)=0] \le \sigma(-\mu_0) + \sigma'(\mu_0-\delta)\delta + \tilde{\epsilon}$$

and

$$\mathbb{E}[Z|\ell(x)=0] \ge (\sigma(-\mu_0) - \sigma'(\mu_0-\delta)\delta)(1-\tilde{\epsilon}),$$

i.e.

$$\mathbb{E}[Z|\ell(x)=0] - \sigma(-\mu_0) \le \sigma'(\mu_0-\delta)\delta + \tilde{\epsilon}$$

and

$$\begin{aligned}\mathbb{E}[Z|\ell(x)=0] - \sigma(-\mu_0) &\ge -\sigma'(\mu_0-\delta)\delta - \tilde{\epsilon}(\sigma(-\mu_0) - \sigma'(\mu_0-\delta)\delta) \\ &= -\sigma'(\mu_0-\delta)\delta(1-\tilde{\epsilon}) - \tilde{\epsilon}\sigma(-\mu_0) \\ &\ge -\sigma'(\mu_0-\delta)\delta - \tilde{\epsilon}.\end{aligned}$$

So

$$\left|\mathbb{E}[Z|\ell(x)=0] - \sigma(-\mu_0)\right| \le \sigma'(\mu_0-\delta)\delta + \tilde{\epsilon}.$$

Similarly

$$\left|\mathbb{E}[Z|\ell(x)=1] - \sigma(\mu_1)\right| \le \sigma'(\mu_1+\delta)\delta + \tilde{\epsilon}.$$

By triangle inequality,

$$\begin{aligned}\left|\sigma(-\mu_0) - \sigma(\mu_1)\right| &\le \left|\sigma(-\mu_0) - \mathbb{E}[Z|\ell(x)=0]\right| + \left|\mathbb{E}[Z|\ell(x)=0] - \mathbb{E}[Z|\ell(x)=1]\right| \\ &\quad + \left|\mathbb{E}[Z|\ell(x)=1] - \sigma(\mu_1)\right| \\ &\le \sigma'(\mu_0-\delta)\delta + \sigma'(\mu_1+\delta)\delta + 3\tilde{\epsilon}.\end{aligned}$$

□

From the proof above, we can directly obtain the following corollary about Z.

Corollary B.3.

$$\mathbb{P}[\left|Z - \mathbb{E}[Z|\ell(x)=0]\right| \le 2\sigma'(\mu_0-\delta)\delta + \tilde{\epsilon}|\ell(x)=0] \ge 1-\tilde{\epsilon}$$
$$\mathbb{P}[\left|Z - \mathbb{E}[Z|\ell(x)=1]\right| \le 2\sigma'(\mu_1+\delta)\delta + \tilde{\epsilon}|\ell(x)=1] \ge 1-\tilde{\epsilon}.$$

In order to obtain the concentration of Z, we need to estimate $\sigma'(\mu_0-\delta)\delta$ and $\sigma'(\mu_1+\delta)\delta$. The following proposition is based on the condition that $|\mu_0+\mu_1| \ge 4\delta$. If $|\mu_0+\mu_1| < 4\delta$, set $c := \mu_0 - \mu_1$, we have $\mu_0 > \frac{c}{2} - 2\delta$ and $\mu_1 < -\frac{c}{2} + 2\delta$. Then the concentration of output shown in Corollary A.4 can guarantee $1-\epsilon$ accuracy of the model for any $\epsilon > 0$. In fact, from $|\Delta - \mu_0| < \delta$, we have $\Delta > \mu_0 - \delta > \frac{c}{2} - 3\delta > 0$, due to $\delta = o(c)$. So

$$\mathbb{P}[\Delta > 0|\ell(x)=0] \ge \mathbb{P}[|\Delta-\mu_0| < \delta|\ell(x)=0] \ge 1-\tilde{\epsilon}.$$

Similarly,

$$\mathbb{P}[\Delta < 0|\ell(x)=1] \ge \mathbb{P}[|\Delta-\mu_0| < \delta|\ell(x)=1] \ge 1-\tilde{\epsilon}.$$

Since $\tilde{\epsilon} = o(\epsilon)$, the model achieves overall accuracy $\ge 1-\epsilon$.

Proposition B.4. *If $|\mu_0+\mu_1| \ge 4\delta$, then $\sigma'(\mu_0-\delta)\delta = O(\tilde{\epsilon}), \sigma'(\mu_1+\delta)\delta = O(\tilde{\epsilon})$.*

Proof. First, we estimate the lower bound of $|\sigma(-\mu_0) - \sigma(\mu_1)|$ via the Fundamental Theorem of Calculus. We have $|\sigma(-\mu_0) - \sigma(\mu_1)| = \left|\int_{-\mu_0}^{\mu_1} \sigma'(t)\,dt\right|$.

If $-\mu_0 < \mu_1 < 0$, since $\mu_0 + \mu_1 \ge 4\delta$, we divide the interval $[-\mu_0, \mu_1]$ into $[-\mu_0, -\mu_0+2\delta] \cup [-\mu_0+2\delta, \mu_1-2\delta] \cup [\mu_1-2\delta, \mu_1]$ and estimate the lower bound of the integral. Since $\sigma'(x)$ is increasing on $(-\infty, 0]$, we have

$$\int_{-\mu_0}^{\mu_1} \sigma'(t)\,dt \ge \sigma'(-\mu_0)\cdot 2\delta + I_1 + \sigma'(\mu_1-2\delta)\cdot 2\delta, \tag{8}$$

where $I_1 = \int_{-\mu_0+2\delta}^{\mu_1-2\delta} \sigma'(t)\,dt$. If $\mu_1 < -\mu_0 < 0$, similarly we have

$$\int_{\mu_1}^{-\mu_0} \sigma'(t)\,dt \geq \sigma'(\mu_1)\cdot 2\delta + I_2 + \sigma'(-\mu_0-2\delta)\cdot 2\delta, \tag{9}$$

where $I_2 = \int_{\mu_1+2\delta}^{-\mu_0-2\delta} \sigma'(t)\,dt$. We have a uniform lower bound from (8) and (9):

$$\left|\int_{-\mu_0}^{\mu_1} \sigma'(t)\,dt\right| \geq \sigma'(-\mu_0-2\delta)\cdot 2\delta + \sigma'(\mu_1-2\delta)\cdot 2\delta + I, \tag{10}$$

where $I = \min\{I_1, I_2\}$.

Furthermore, by Proposition B.2,

$$|\sigma(-\mu_0) - \sigma(\mu_1)| \leq \sigma'(\mu_0-\delta)\delta + \sigma'(\mu_1+\delta)\delta + 3\tilde{\epsilon}. \tag{11}$$

Combine (10) and (11):

$$2\sigma'(-\mu_0-2\delta)\delta + 2\sigma'(\mu_1-2\delta)\delta \leq \sigma'(-\mu_0+\delta)\delta + \sigma'(\mu_1+\delta)\delta + 3\tilde{\epsilon}. \tag{12}$$

By Lagrange mean value theorem,

$$\sigma'(-\mu_0-2\delta) = \sigma'(-\mu_0+\delta) - 3\sigma''(\xi_0)\delta$$
$$\sigma'(\mu_1-2\delta) = \sigma'(\mu_1+\delta) - 3\sigma''(\xi_1)\delta,$$

where $\xi_0 \in (-\mu_0-2\delta, -\mu_0+\delta), \xi_1 \in (\mu_1-2\delta, \mu_1+\delta)$. Plug these into (12):

$$\sigma'(\mu_0-\delta)\delta + \sigma'(\mu_1+\delta)\delta - 6\delta^2(\sigma''(\xi_0)+\sigma''(\xi_1)) \leq 3\tilde{\epsilon}.$$

Since $\delta^2(\sigma''(\xi_0)+\sigma''(\xi_1)) = o(\sigma'(\mu_0-\delta)\delta)$ and $o(\sigma'(\mu_1+\delta)\delta)$, we have

$$\sigma'(\mu_0-\delta)\delta = O(\tilde{\epsilon})$$
$$\sigma'(\mu_1+\delta)\delta = O(\tilde{\epsilon}).$$

□

Combine Proposition B.4 and Corollary B.3, we have the following concentration of Z.

Proposition B.5.

$$\mathbb{P}[\big|Z - \mathbb{E}[Z|\ell(x)=0]\big| \leq O(\tilde{\epsilon})|\ell(x)=0] \geq 1-\tilde{\epsilon}$$
$$\mathbb{P}[\big|Z - \mathbb{E}[Z|\ell(x)=1]\big| \leq O(\tilde{\epsilon})|\ell(x)=1] \geq 1-\tilde{\epsilon}.$$

Under the condition of balanced performance, we have the following corollary about the concentration of Z independent of the label of x.

Corollary B.6. *If* $\big|\mathbb{E}[\frac{\partial L}{\partial b_0}]\big| \leq \frac{\tilde{\epsilon}}{2}$, *then* $\mathbb{P}[\big|Z - \mathbb{E}[Z]\big| \leq O(\tilde{\epsilon})] \geq 1-\tilde{\epsilon}$.

Proof. Since $\mathbb{E}[\frac{\partial L}{\partial b_0}] = \frac{1}{2}(\mathbb{E}[Z|\ell(x)=0] - \mathbb{E}[Z|\ell(x)=1])$, we have

$$\big|\mathbb{E}[Z|\ell(x)=0] - \mathbb{E}[Z|\ell(x)=1]\big| \leq \tilde{\epsilon}.$$

On the other hand,

$$\mathbb{E}[Z] = \frac{1}{2}\big(\mathbb{E}[Z|\ell(x)=0] + \mathbb{E}[Z|\ell(x)=1]\big).$$

So we have $\big|\mathbb{E}[Z] - \mathbb{E}[Z|\ell(x)=0]\big| \leq \frac{\tilde{\epsilon}}{2}$. By Proposition B.5, $\mathbb{P}[|Z-\mathbb{E}[Z]| \leq O(\tilde{\epsilon})] \geq 1-\tilde{\epsilon}$. □

Now we can derive the estimation of the derivatives.

Theorem B.7 (concentration of derivatives)**.** *For loss on the whole graph $L = L(G)$, with probability $\geq 1 - O(\frac{1}{n})$, we have* [3]

[3] Since $\frac{\partial L}{\partial \beta_i'} = -\frac{\partial L}{\partial \beta_i}$ (see Theorem B.1), we only need to estimate $\frac{\partial L}{\partial \beta_i}$.

1. *If* $\alpha_i > \alpha_i' > 0$ *or* $\alpha_i > 0 > \alpha_i', |\frac{\alpha_i}{\alpha_i'}| \geq \frac{p}{q}(1+\log^{-\frac{1}{3}} n)$, *then*

$$\left|\frac{\partial L}{\partial \alpha_i} + (\beta_i - \beta_i')\frac{\lambda}{2}\left(\frac{p-q}{p+q}\right)^2 \mathbb{E}[Z]\right| \leq |\beta_i - \beta_i'|\mathbb{E}[Z]O(\log^{-\frac{1}{2}} n) \tag{13}$$

$$\left|\frac{\partial L}{\partial \alpha_i'} - (\beta_i - \beta_i')\frac{\lambda}{2}\left(\frac{p-q}{p+q}\right)^2 \mathbb{E}[Z]\right| \leq |\beta_i - \beta_i'|\mathbb{E}[Z]O(\log^{-\frac{1}{2}} n) \tag{14}$$

$$\left|\frac{\partial L}{\partial \beta_i} + (\alpha_i - \alpha_i')\frac{\lambda}{2}\left(\frac{p-q}{p+q}\right)^2 \mathbb{E}[Z]\right| \leq |\alpha_i - \alpha_i'|\mathbb{E}[Z]O(\log^{-\frac{1}{2}} n). \tag{15}$$

2. *If* $\alpha_i > 0 > \alpha_i', |\frac{\alpha_i}{\alpha_i'}| \in [\frac{q}{p}(1+\gamma), \frac{p}{q}(1-\log^{-\frac{1}{3}} n)]$, *where* $\gamma \in [\log^{-\frac{1}{3}} n, (\frac{p}{q})^2(1-\log^{-\frac{1}{3}} n) - 1]$, *then*

$$\left|\frac{\partial L}{\partial \alpha_i} + (\beta_i - \beta_i')\frac{\lambda p(p-q)}{2(p+q)^2}\mathbb{E}[Z]\right| \leq |\beta_i - \beta_i'|\mathbb{E}[Z]O(\log^{-\frac{1}{2}} n) \tag{16}$$

$$\left|\frac{\partial L}{\partial \alpha_i'} + (\beta_i - \beta_i')\frac{\lambda q(p-q)}{2(p+q)^2}\mathbb{E}[\mathbb{Z}]\right| \leq |\beta_i - \beta_i'|\mathbb{E}[Z]O(\log^{-\frac{1}{2}} n) \tag{17}$$

$$\left|\frac{\partial L}{\partial \beta_i} + \frac{\lambda(p-q)(p\alpha_i + q\alpha_i')}{2(p+q)^2}\mathbb{E}[Z]\right| \leq |\alpha_i - \alpha_i'|\mathbb{E}[Z]O(\log^{-\frac{1}{2}} n). \tag{18}$$

3. *If* $\alpha_i > 0 > \alpha_i', |\frac{\alpha_i}{\alpha_i'}| \in (\frac{p}{q}(1-\log^{-\frac{1}{3}} n), \frac{p}{q}(1+\log^{-\frac{1}{3}} n))$, *then*

$$\frac{\partial L}{\partial \beta_i} \in [-(\alpha_i - \alpha_i')\mathbb{E}[Z]\left(\frac{\lambda(p-q)\Lambda_1}{2(p+q)^2} + O(\log^{-\frac{1}{2}} n)\right), \\ -(\alpha_i - \alpha_i')\mathbb{E}[Z]\left(\frac{\lambda(p-q)(\Lambda_3 - \Lambda_2)}{2(p+q)^2} - O(\log^{-\frac{1}{2}} n)\right)], \tag{19}$$

where $\Lambda_1 = \frac{(1+\log^{-\frac{1}{3}} n)p^2 - q^2}{(1+\log^{-\frac{1}{3}} n)p+q}, \Lambda_2 = \frac{pq\log^{-\frac{1}{3}} n}{(1+\log^{-\frac{1}{3}} n)p+q}$ *and* $\Lambda_3 = \frac{(1-\log^{-\frac{1}{3}} n)p^2 - q^2}{(1-\log^{-\frac{1}{3}} n)p+q}$;

- *if* $\beta_i > \beta_i'$,

$$\frac{\partial L}{\partial \alpha_i} \in [-(\beta_i - \beta_i')\mathbb{E}[Z]\left(\frac{\lambda p(p-q)}{2(p+q)^2} + O(\log^{-\frac{1}{2}} n)\right), \\ -(\beta_i - \beta_i')\mathbb{E}[Z]\left(\frac{\lambda}{2}\left(\frac{p-q}{p+q}\right)^2 - O(\log^{-\frac{1}{2}} n)\right)] \tag{20}$$

$$\frac{\partial L}{\partial \alpha_i'} \in [-(\beta_i - \beta_i')\mathbb{E}[Z]\left(\frac{\lambda q(p-q)}{2(p+q)^2} + O(\log^{-\frac{1}{2}} n)\right), \\ (\beta_i - \beta_i')\mathbb{E}[Z]\left(\frac{\lambda}{2}\left(\frac{p-q}{p+q}\right)^2 + O(\log^{-\frac{1}{2}} n)\right)] \tag{21}$$

$$\frac{\partial L}{\partial \alpha_i} - \frac{\partial L}{\partial \alpha_i'} \in [-(\beta_i - \beta_i')\mathbb{E}[Z]\left(\lambda\left(\frac{p-q}{p+q}\right)^2 + O(\log^{-\frac{1}{2}} n)\right), \\ -(\beta_i - \beta_i')\mathbb{E}[Z]\left(\frac{\lambda}{2}\left(\frac{p-q}{p+q}\right)^2 - O(\log^{-\frac{1}{2}} n)\right)], \tag{22}$$

- *if* $\beta_i \leq \beta_i'$,

$$\frac{\partial L}{\partial \alpha_i} \in [-(\beta_i - \beta_i')\mathbb{E}[Z]\left(\frac{\lambda}{2}\left(\frac{p-q}{p+q}\right)^2 - O(\log^{-\frac{1}{2}} n)\right), \\ -(\beta_i - \beta_i')\mathbb{E}[Z]\left(\frac{\lambda p(p-q)}{2(p+q)^2} + O(\log^{-\frac{1}{2}} n)\right)] \tag{23}$$

$$\frac{\partial L}{\partial \alpha_i'} \in [-(\beta_i - \beta_i')\mathbb{E}[Z]\left(\frac{\lambda}{2}\left(\frac{p-q}{p+q}\right)^2 + O(\log^{-\frac{1}{2}} n)\right), \\ (\beta_i - \beta_i')\mathbb{E}[Z]\left(\frac{\lambda q(p-q)}{2(p+q)^2} + O(\log^{-\frac{1}{2}} n)\right)] \quad (24)$$

$$\frac{\partial L}{\partial \alpha_i} - \frac{\partial L}{\partial \alpha_i'} \in [-(\beta_i - \beta_i')\mathbb{E}[Z]\left(\frac{\lambda}{2}\left(\frac{p-q}{p+q}\right)^2 - O(\log^{-\frac{1}{2}} n)\right), \\ -(\beta_i - \beta_i')\mathbb{E}[Z]\left(\lambda\left(\frac{p-q}{p+q}\right)^2 + O(\log^{-\frac{1}{2}} n)\right)] \quad (25)$$

4. *If* $\alpha_i > 0 > \alpha_i', \left|\frac{\alpha_i}{\alpha_i'}\right| \le \frac{q}{p}(1 + \log^{-\frac{1}{3}} n), \beta_i < \beta_i'$, *then*

$$\frac{\partial L}{\partial \alpha_i} - \frac{\partial L}{\partial \alpha_i'} \ge -|\beta_i - \beta_i'|O(\tilde{\epsilon}) \quad (26)$$

$$\frac{\partial L}{\partial \beta_i} \le O(\tilde{\epsilon}). \quad (27)$$

Proof. We show the proof for item 1, other items can be proved similarly. Since $L(G)$ is the average of the losses over revealed vertices, we first show the concentration of $\frac{\partial L(x)}{\partial \alpha_i}$, then we show the concentration of $\frac{\partial L(G)}{\partial \alpha_i}$ using union bound. Since

$$\frac{\partial L(x)}{\partial \alpha_i} = (-1)^{1-\ell(x)} 4(\beta_i - \beta_i')Z\frac{1}{n}\sum_y \frac{\mathbb{1}[y \sim x]\mathbb{1}[\alpha_i t_0^y + \alpha_i' t_1^y \ge 0]t_0^y}{n(p+q)^2},$$

we first show the concentration of $\sum_y \frac{\mathbb{1}[y\sim x]\mathbb{1}[\alpha_i t_0^y + \alpha_i' t_1^y \ge 0]t_0^y}{n(p+q)^2}$ using Chernoff bound. Similar as the proof of Theorem A.1, the summands are conditionally independent. Set $U := \frac{\mathbb{1}[y\sim x]\mathbb{1}[\alpha_i t_0^y + \alpha_i' t_1^y \ge 0]t_0^y}{n(p+q)^2}$, let A_0 and A_1 denote the events:

$$A_0 : \left|t_0^y - \frac{n\lambda}{2}p\right| \le O(n^{\frac{2}{3}}) \text{ and } \left|t_1^y - \frac{n\lambda}{2}q\right| \le O(n^{\frac{2}{3}}),$$
$$A_1 : \left|t_0^y - \frac{n\lambda}{2}q\right| \le O(n^{\frac{2}{3}}) \text{ and } \left|t_1^y - \frac{n\lambda}{2}p\right| \le O(n^{\frac{2}{3}}).$$

By tail bound of binomial random variables:

$$\mathbb{P}[A_0|\ell(y) = 0] \ge 1 - O(\tilde{\epsilon})$$
$$\mathbb{P}[A_1|\ell(y) = 1] \ge 1 - O(\tilde{\epsilon}).$$

For x with $\ell(x) = 0$, we define[4]

$$\mu := \mathbb{E}[U|0,0] = \mathbb{E}[U|0,0,A_0]\mathbb{P}[A_0|0] + \mathbb{E}[U|0,0,\bar{A}_0]\mathbb{P}[\bar{A}_0|0].$$

By the definition of A_0,

$$\mu \ge \frac{1}{n(p+q)^2}(p(\frac{n\lambda}{2}p - O(n^{\frac{2}{3}})))(1 - O(\tilde{\epsilon})) \ge \frac{\lambda}{2}\left(\frac{p}{p+q}\right)^2 - O(n^{-\frac{1}{3}+\tau})$$
$$\mu \le \frac{1}{n(p+q)^2}(p(\frac{n\lambda}{2}p + O(n^{\frac{2}{3}})) + O(\tilde{\epsilon})) \le \frac{\lambda}{2}\left(\frac{p}{p+q}\right)^2 + O(n^{-\frac{1}{3}+\tau}).$$

By Chernoff bound

$$\mathbb{P}[\left|\sum_{\ell(y)=0} U - \frac{n\lambda}{4}\left(\frac{p}{p+q}\right)^2\right| \le O(n^{\frac{2}{3}+\tau})] \ge 1 - \exp(-\Omega(n^{\frac{2}{3}-4\tau})).$$

[4]We write $\mathbb{E}[U|\ell(x) = 0, \ell(y) = 0]$ as $\mathbb{E}[U|0,0]$, $\mathbb{P}[A_0|\ell(y) = 0]$ as $\mathbb{P}[A_0|0]$ for brevity, other terms are similar.

Similarly

$$\mathbb{P}[\left|\sum_{\ell(y)=1} U - \frac{n\lambda}{4}\left(\frac{q}{p+q}\right)^2\right| \leq O(n^{\frac{2}{3}+\tau})] \geq 1 - \exp\left(-\Omega(n^{\frac{2}{3}-4\tau})\right).$$

Hence

$$\mathbb{P}[\left|\sum_{y} U - \frac{n\lambda}{4}\frac{p^2+q^2}{(p+q)^2}\right| \leq O(n^{\frac{2}{3}+\tau})] \geq 1 - \exp\left(-\Omega(n^{\frac{2}{3}-4\tau})\right).$$

By Corollary B.6, $\mathbb{P}[|Z - \mathbb{E}[Z]| \leq O(\tilde{\epsilon})] \geq 1 - \tilde{\epsilon}$, so we have

$$\mathbb{P}[\left|\frac{\partial L(x)}{\partial \alpha_i} + (\beta_i - \beta_i')\lambda\frac{p^2+q^2}{(p+q)^2}\mathbb{E}[Z]\right| \leq |\beta_i - \beta_i'|\mathbb{E}[Z]O(n^{-\frac{1}{3}+\tau})|\ell(x) = 0] \geq 1 - \exp\left(-\Omega(n^{\frac{2}{3}-4\tau})\right).$$

For $\ell(x) = 1$, similarly we have

$$\mathbb{P}[\left|\frac{\partial L(x)}{\partial \alpha_i} - (\beta_i - \beta_i')\lambda\frac{2pq}{(p+q)^2}\mathbb{E}[Z]\right| \leq |\beta_i - \beta_i'|\mathbb{E}[Z]O(n^{-\frac{1}{3}+\tau})|\ell(x) = 1] \geq 1 - \exp\left(-\Omega(n^{\frac{2}{3}-4\tau})\right).$$

By union bound, we have (13). (14) and (15) can be proved similarly. □

Using Theorem B.7, we can analyze dynamics of neurons of each type. First, we introduce some notations. Let η_k denote the learning rate at the k-th epoch, $Z^{(k)}$ be the value of Z at the k-th epoch, $\alpha_i^{(k)}$ be the value of α_i at the k-th epoch, similar for $\alpha_i'^{(k)}, \beta_i^{(k)}$ and $\beta_i'^{(k)}$. In particular, $\alpha_i^{(0)}, \alpha_i'^{(0)}, \beta_i^{(0)}$ and $\beta_i'^{(0)}$ represent the values at initialization.

B.1 "Good Type" Neurons

In this section, we show that "good type" neurons stay in the "good type" regime throughout coordinate descent (Theorem B.8) using Theorem B.7.

Theorem B.8. *"Good type" neurons are preserved in the "good type" throughout coordinate descent with probability* $\geq 1 - \exp\left(-\Omega(n^{\frac{2}{3}-4\tau})\right)$ *over the SBM randomness.*

Proof. As shown in section 3.2, "good type" regime is composed of equation G_1 and equation G_2, we show the dynamics of neurons in equation G_1 and equation G_2 respectively.

Assume that neuron $(\alpha_i^{(k)}, \alpha_i'^{(k)}, \beta_i^{(k)}, \beta_i'^{(k)})$ is in equation G_1, we show that it either stays in equation G_1 or moves into equation G_2 throughout coordinate descent. In fact, by (15), with probability $\geq 1 - \exp\left(-\Omega(n^{\frac{2}{3}-4\tau})\right)$, $\frac{\partial L}{\partial \beta_i^{(k)}} < 0 < \frac{\partial L}{\partial \beta_i'^{(k)}}$, so $\beta_i^{(k+1)} > \beta_i^{(k)}, \beta_i'^{(k+1)} < \beta_i'^{(k)}$ and hence $\beta_i^{(k+1)} - \beta_i'^{(k+1)} > \beta_i^{(k)} - \beta_i'^{(k)} > 0$. By (13) and (14), $\frac{\partial L}{\partial \alpha_i^{(k)}} < 0 < \frac{\partial L}{\partial \alpha_i'^{(k)}}$, so $\alpha_i^{(k+1)} > \alpha_i^{(k)}, \alpha_i'^{(k+1)} < \alpha_i'^{(k)}$. If $\alpha_i'^{(k+1)} > 0$, this neuron stays in equation G_1. If $\alpha_i'^{(k+1)} < 0$, since

$$\left|\frac{\alpha_i^{(k+1)}}{\alpha_i'^{(k+1)}}\right| = \left|\frac{\alpha_i^{(k)} - \eta_k\frac{\partial L}{\partial \alpha_i^{(k)}}}{\alpha_i'^{(k)} - \eta_k\frac{\partial L}{\partial \alpha_i'^{(k)}}}\right| > 1,$$

the neuron moves into equation G_2.

Assume that neuron is in equation G_2, we also show that it either moves into equation G_1 or stays in equation G_2. As shown in section 3.2, equation $G_2 =$ equation $G_{2,1} \cup$ equation $G_{2,2} \cup$ equation $G_{2,3}$. If the neuron is in equation $G_{2,1}$, again by (13), (14) and (15), $\alpha_i^{(k+1)} > \alpha_i^{(k)} > 0 > \alpha_i'^{(k)} > \alpha_i'^{(k+1)}, \left|\frac{\alpha_i^{(k+1)}}{\alpha_i'^{(k+1)}}\right| > 1, \beta_i^{(k+1)} > \beta_i^{(k)} > \beta_i'^{(k)} > \beta_i'^{(k+1)}$, so the neuron stays in equation $G_{2,2}$. If the neuron is in equation $G_{2,2}$, by (16), (17) and (18), $\frac{\partial L}{\partial \beta_i^{(k)}} < 0 < \frac{\partial L}{\partial \beta_i'^{(k)}}$, so

$\beta_i^{(k+1)} > \beta_i'^{(k+1)}$. Also, $\frac{\partial L}{\partial \alpha_i^{(k)}} < \frac{\partial L}{\partial \alpha_i'^{(k)}} < 0$, so $\alpha_i^{(k+1)} > \alpha_i'^{(k+1)}, \left|\frac{\alpha_i^{(k+1)}}{\alpha_i'^{(k+1)}}\right| > \left|\frac{\alpha_i^{(k)}}{\alpha_i'^{(k)}}\right| > 1$. If $\alpha_i'^{(k+1)} < 0$, the neuron stays in equation G_2. If $\alpha_i'^{(k+1)} > 0$, it moves into equation G_1. If the neuron is in equation $G_{2,3}$, by (19) and (22), $\frac{\partial L}{\partial \beta_i^{(k)}} < 0 < \frac{\partial L}{\partial \beta_i'^{(k)}}, \frac{\partial L}{\partial \alpha_i^{(k)}} - \frac{\partial L}{\partial \alpha_i'^{(k)}} < 0$, so $\beta_i^{(k+1)} > \beta_i^{(k)} > \beta_i'^{(k)} > \beta_i'^{(k+1)}, \alpha_i^{(k+1)} - \alpha_i'^{(k+1)} > \alpha_i^{(k)} - \alpha_i'^{(k)} > 0$. By (20) and (21),

$$\frac{\partial L}{\partial \alpha_i} \leq -(\beta_i - \beta_i')\mathbb{E}[Z]\left(\frac{\lambda}{2}\left(\frac{p-q}{p+q}\right)^2 - O(n^{-\frac{1}{3}})\right),$$
$$\frac{\partial L}{\partial \alpha_i'} \leq (\beta_i - \beta_i')\mathbb{E}[Z]\left(\frac{\lambda}{2}\left(\frac{p-q}{p+q}\right)^2 + O(n^{-\frac{1}{3}})\right).$$

Similar as in equation $G_{2,2}$, if $\alpha_i'^{(k+1)} < 0$, the neuron stays in equation G_2. If $\alpha_i'^{(k+1)} > 0$, it moves into equation G_1. □

B.2 "Bad Type" Neurons

As shown in Section 5, neurons of "bad type" consist of two cases: equation B_1 and equation B_2, where equation $B_2 =$ equation $B_{2,1} \cup$ equation $B_{2,2} \cup$ equation $B_{2,3} \cup$ equation B_3. Since the output in equation B_3 is concentrated at 0 (see Theorem A.2), we don't need to worry if neurons move into this region. Neurons in equation $B_1 \cup$ equation $B_{2,1} \cup$ equation $B_{2,2} \cup$ equation $B_{2,3}$ might exit "bad type" regime and become "harmless" or "good" (if the neuron becomes order-aligned), which will do no harm to the performance of the model. If they stay in equation $B_1 \cup$ equation $B_{2,1}$ $\cup$ equation $B_{2,2} \cup$ equation $B_{2,3}$, the following theorem shows that the separation $m_0^i - m_1^i$ can be upper bounded by initialization. In fact, Theorem 5.3 shows that $m_0^i - m_1^i$ is proportional to $|\alpha_i - \alpha_i'||\beta_i - \beta_i'|$. The next theorem shows that both $|\alpha_i - \alpha_i'|$ and $|\beta_i - \beta_i'|$ shrink throughout coordinate descent. The worst situation is that the magnitude of $|\alpha_i - \alpha_i'|$ and $|\beta_i - \beta_i'|$ of neurons in equation B_3 increase and move into equation B_1 or equation B_2 at certain epoch. From Theorem B.7 we see that the magnitude can only increase by a limited rate (we can see this more explicitly in Theorem C.2).

Theorem B.9. *If $(\alpha_i^{(k)}, \alpha_i'^{(k)}, \beta_i^{(k)}, \beta_i'^{(k)})$ is in equation $B_1 \cup$ equation $B_{2,1} \cup$ equation $B_{2,2} \cup$ equation $B_{2,3}$ then with probability $\geq 1 - \exp\left(-\Omega(n^{\frac{2}{3}-4\tau})\right)$ over the SBM randomness, $\left|\alpha_i^{(k+1)} - \alpha_i'^{(k+1)}\right| \leq \left|\alpha_i^{(k)} - \alpha_i'^{(k)}\right|$, $\left|\beta_i^{(k+1)} - \beta_i'^{(k+1)}\right| \leq \left|\beta_i^{(k)} - \beta_i'^{(k)}\right|$.*

Proof. In equation B_1 and equation $B_{2,1}$, by (13) and (14), $\frac{\partial L}{\partial \alpha_i^{(k)}} > 0 > \frac{\partial L}{\partial \alpha_i'^{(k)}}$, then $\alpha_i^{(k+1)} < \alpha_i^{(k)}, \alpha_i'^{(k+1)} > \alpha_i'^{(k)}$, so $|\alpha_i^{(k+1)} - \alpha_i'^{(k+1)}| \leq |\alpha_i^{(k)} - \alpha_i'^{(k)}|$. Similarly, by (15), $\frac{\partial L}{\partial \beta_i^{(k)}} = -\frac{\partial L}{\partial \beta_i'^{(k)}} < 0$, so $|\beta_i^{(k+1)} - \beta_i'^{(k+1)}| \leq |\beta_i^{(k)} - \beta_i'^{(k)}|$ (Note that $\alpha_i^{(k)} > \alpha_i'^{(k)}, \beta_i^{(k)} < \beta_i'^{(k)}$).

In equation $B_{2,2}$, from (16) and (17), we have $\frac{\partial L}{\partial \alpha_i^{(k)}} > \frac{\partial L}{\partial \alpha_i'^{(k)}} > 0$, so $|\alpha_i^{(k+1)} - \alpha_i'^{(k+1)}| \leq |\alpha_i^{(k)} - \alpha_i'^{(k)}|$. On the other hand, $\frac{\partial L}{\partial \beta_i^{(k)}} < 0 < \frac{\partial L}{\partial \beta_i'^{(k)}}$, so $\beta_i^{(k+1)} > \beta_i^{(k)}, \beta_i'^{(k+1)} < \beta_i'^{(k)}$ and $|\beta_i^{(k+1)} - \beta_i'^{(k+1)}| \leq |\beta_i^{(k)} - \beta_i'^{(k)}|$. In equation $B_{2,3}$, by (25), $\frac{\partial L}{\partial \alpha_i^{(k)}} - \frac{\partial L}{\partial \alpha_i'^{(k)}} > 0$, so $|\alpha_i^{(k+1)} - \alpha_i'^{(k+1)}| \leq |\alpha_i^{(k)} - \alpha_i'^{(k)}|$. By (19), $\frac{\partial L}{\partial \beta_i^{(k)}} < 0 < \frac{\partial L}{\partial \beta_i'^{(k)}}$, so $|\beta_i^{(k+1)} - \beta_i'^{(k+1)}| \leq |\beta_i^{(k)} - \beta_i'^{(k)}|$. □

B.3 "Harmless Type" Neurons

Section 5 shows that there are two cases of "harmless type": equation H_1 and equation H_2. For neurons in equation H_1, the derivatives of parameters are estimated in (16), (17) and (18) (same as in equation $G_{2,2}$). We can have similar analysis as in equation $G_{2,2}$ and show that the inequality $\alpha_i > 0 > \alpha_i', \beta_i > \beta_i'$ can be preserved. Moreover $\left|\frac{\alpha_i}{\alpha_i'}\right|$ increases. So the neurons either stay in equation H_1 or become "good type" if $\left|\frac{\alpha_i}{\alpha_i'}\right| > 1$. In particular, neurons in equation H_1 do no harm to the performance of the model.

For neurons in equation H_2, $\mathbb{1}[\alpha_i t_0^y + \alpha_i' t_1^y \geq 0] = 0$, so the derivatives are all equal to 0. Therefore they are never updated. Meanwhile they don't affect the performance of the model since $\phi(\alpha_i t_0^y + \alpha_i' t_1^y) = 0$ and $\Delta_i = 0$.

C LEARNING GUARANTEE

In this section, we show that, with high probability GCN can detect communities in SBM with desired accuracy. The following theorem shows that if μ_0 and μ_1 are separated enough, then the model achieves high accuracy.

Theorem C.1. *$\forall \epsilon > 0$, if $\left|\mathbb{E}[\frac{\partial L}{\partial b_0}]\right| < \frac{\epsilon}{4}$ and $\sigma(-\frac{\mu_0 - \mu_1}{2}) < \frac{\epsilon}{2}$, then $\mathbb{P}[\Delta < 0 | \ell(x) = 0] < 4\epsilon, \mathbb{P}[\Delta > 0 | \ell(x) = 1] < 4\epsilon$.*

Proof. We prove by contradiction. Suppose

$$\mathbb{P}[\Delta < 0 | \ell(x) = 0] \geq 4\epsilon, \tag{28}$$

then

$$\begin{aligned}\mathbb{E}[Z|\ell(x) = 0] &= \mathbb{E}[Z|\ell(x) = 0, \Delta < 0]\mathbb{P}[\Delta < 0|\ell(x) = 0] \\ &\quad + \mathbb{E}[Z|\ell(x) = 0, \Delta \geq 0]\mathbb{P}[\Delta \geq 0|\ell(x) = 0] \\ &\geq \frac{1}{2} \cdot 4\epsilon = 2\epsilon. \end{aligned} \tag{29}$$

Furthermore, we claim that $\mu_0 < \delta$. In fact, if $\mu_0 \geq \delta$, since $\mathbb{P}[|\Delta - \mu_0| \leq \delta | \ell(x) = 0] \geq 1 - \epsilon$ by Corollary A.4, and $\Delta \geq \mu_0 - \delta \geq 0$, we have

$$\mathbb{P}[\Delta \geq 0 | \ell(x) = 0] \geq \mathbb{P}[|\Delta - \mu_0| \leq \delta | \ell(x) = 0] \geq 1 - \epsilon,$$

i.e. $\mathbb{P}[\Delta < 0 | \ell(x) = 0] \leq \epsilon$, which contradicts (28).

Let $c := \mu_0 - \mu_1$, then $\mu_1 = \mu_0 - c < \delta - c$. Again, by Corollary A.4, for $\ell(x) = 1, \Delta < \mu_1 + \delta$ with probability $\geq 1 - \epsilon$, we have $Z = \sigma(\Delta) < \sigma(\mu_1 + \delta) < \sigma(-c + 2\delta)$. Then

$$\begin{aligned}\mathbb{E}[Z|\ell(x) = 1] &= \mathbb{E}[Z|\ell(x) = 1, |\Delta - \mu_1| < \delta]\mathbb{P}[|\Delta - \mu_1| < \delta | \ell(x) = 1] \\ &\quad + \mathbb{E}[Z|\ell(x) = 1, |\Delta - \mu_1| \geq \delta]\mathbb{P}[|\Delta - \mu_1| \geq \delta | \ell(x) = 1] \\ &< \sigma(-c + 2\delta) \cdot 1 + 1 \cdot \epsilon \\ &< \sigma\big(-\frac{c}{2}\big) + \epsilon. \end{aligned}$$

The last step is due to $\delta = o(c)$. Since $\sigma(-\frac{c}{2}) < \frac{\epsilon}{2}, \mathbb{E}[Z|\ell(x) = 1] < \frac{3\epsilon}{2}$. Combine with (29),

$$\left|\mathbb{E}[Z|\ell(x) = 0] - \mathbb{E}[Z|\ell(x) = 1]\right| > \frac{\epsilon}{2}.$$

On the other hand, $\left|\mathbb{E}[\frac{\partial L}{\partial b_0}]\right| < \frac{\epsilon}{4}$ implies

$$\left|\mathbb{E}[Z|\ell(x) = 0] - \mathbb{E}[Z|\ell(x) = 1]\right| < \frac{\epsilon}{2},$$

which is a contradiction. So $\mathbb{P}[\Delta < 0 | \ell(x) = 0] < 4\epsilon$. Similarly, $\mathbb{P}[\Delta > 0 | \ell(x) = 1] < 4\epsilon$. □

From this theorem, we see that if $\mu_0 - \mu_1 \geq 2 \log \frac{2}{\epsilon}$, the accuracy $\geq 1 - 4\epsilon$. In the following, we show that the model can achieve such separation between μ_0 and μ_1 through coordinate descent. In order to make constant update of parameters at every epoch, we set an adaptive learning rate $\eta_k = \frac{1}{\mathbb{E}[Z^{(k)}]}$. We first refine Corollary 5.3 about the separation of output for each type of neuron $(m_0^i - m_1^i)$ using the dynamics of parameters.

Theorem C.2 (separation of output)**.** *Let m_0^i and m_1^i be defined as in Section 5, train the model for k epochs by the defined coordinate descent with adaptive learning rate $\eta_k = \frac{1}{\mathbb{E}[Z^{(k)}]}$, with probability $\geq 1 - \exp\left(-\Omega(n^{\frac{2}{3} - 4\tau})\right)$ over the SBM randomness,*

- *if the i-th neuron is of "good type", then*

$$m_0^i - m_1^i \geq A_i^{(0)} B_i^{(0)} \frac{\lambda}{2} \left(\frac{p-q}{p+q} \right)^2 \left(1 + \frac{\sqrt{2}\lambda}{8} \left(\frac{p-q}{p+q} \right)^2 \right)^{2k},$$

- *if the i-th neuron is of "bad type", then*

$$m_0^i - m_1^i \geq -k\big(\big(A_i^{(0)}\big)^2 + \big(B_i^{(0)}\big)^2 \big) \frac{\lambda p(p-q)}{(p+q)^2},$$

- *if the i-th neuron is of "harmless type", then*

$$m_0^i - m_1^i \geq 0,$$

where $A_i^{(0)} = \alpha_i^{(0)} - \alpha_i'^{(0)}, B_i^{(0)} = \beta_i^{(0)} - \beta_i'^{(0)}$.