Breaking the Curse of Depth in Graph Convolutional Networks via Refined Initialization Strategy

Senmiao Wang [*1] **Yupeng Chen** [*2] **Yushun Zhang** [3] **Tian Ding** [4] **Ruoyu Sun** [3]

Abstract

Graph convolutional networks (GCNs) suffer from the *curse of depth*, a phenomenon where performance degrades significantly as network depth increases. While over-smoothing has been considered the primary cause of this issue, we discover that gradient vanishing or exploding under commonly-used initialization methods also contributes to the curse of depth. To this end, we propose to evaluate GCN initialization quality from three aspects: forward-propagation, backward-propagation, and output diversity. We theoretically prove that conventional initialization methods fail to simultaneously maintain reasonable forward propagation and output diversity. To tackle this problem, We develop a new GCN initialization method called **S**ignal **P**ropagation **o**n **G**raph (SPoGInit). By carefully designing and optimizing initial weight metrics, SPoGInit effectively alleviates performance degradation in deep GCNs. We further introduce a new architecture termed ReZeroGCN, which simultaneously addresses the three aspects at initialization. This architecture achieves performance gains on node classification tasks when increasing the depth from 4 to 64, e.g., 10% gain in training and 3% gain in test accuracy on OGBN-Arxiv. To the best of our knowledge, this is the first result to fully resolve the curse of depth on OGBN-Arxiv over such a range of depths.

1. Introduction

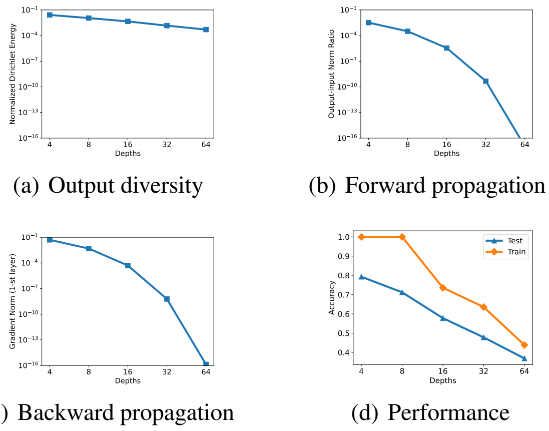
Deep neural networks (DNNs) have consistently shown remarkable success across various domains, with their performance often improving with the increase in depth. For instance, the VGG16 network (Simonyan & Zisserman, 2015), which expanded AlexNet's (Krizhevsky et al., 2012) 8-layer architecture to 16 layers, exhibited a significant boost in test accuracy from 63.3% to 74.4% on ImageNet (Deng et al., 2009). This trend continued with ResNet (He et al., 2016), which achieved 78.57% test accuracy by increasing the network depth to 152 layers. However, in the realm of Graph Convolutional Networks (GCNs) (Wu et al., 2020), deepening the network doesn't always yield similar benefits, and can potentially deteriorate performance. This phenomenon, which we refer to as the *curse of depth*, poses a major challenge in the development of effective GCNs.

In recent years, over-smoothing (Li et al., 2018; Oono & Suzuki, 2019) has been identified as one of the major reasons behind the curse of depth. Over-smoothing occurs when, as a GCN becomes deeper, embeddings among different nodes become increasingly similar, rendering nodes challenging to differentiate. This phenomenon is particularly harmful to GCNs in node classification tasks, where the objective is to assign labels or categories to nodes in a graph based on their features and the graph topology.

A variety of approaches have been explored to tackle the over-smoothing issue within the GCN family, such as nodes or edges dropping techniques (Srivastava et al., 2014; Zou et al., 2019; Rong et al., 2020; Huang et al., 2020; Lu et al., 2021), normalization techniques (Ioffe & Szegedy, 2015; Zhao & Akoglu, 2020; Zhou et al., 2020; Yang et al., 2020; Zhou et al., 2021a; Li et al., 2020; Guo et al., 2023), and regularization techniques (Chen et al., 2020a; Yang et al., 2020; Zhou et al., 2021b). Despite their good performance, they have not fully alleviated the curse of depth. In fact, the optimal performance for GCNs in most of these studies is still achieved with less than 20 layers, suggesting that the curse of depth continues to constrain the potential of GCN. (We summarize the optimal performance and the corresponding depths in Appendix E.) Such limitation signifies an ongoing need for new perspectives and strategies to fully resolve the curse of depth.

[*] Equal contribution. Work done while visiting Prof. Ruoyu Sun. [1] Department of Industrial Engineering and Management Sciences, Northwestern University [2] College of Mathematics, Sichuan University [3] School of Data Science, The Chinese University of Hong Kong, Shenzhen [4] Department of Information Engineering, The Chinese University of Hong Kong. Correspondence to: Ruoyu Sun <sunruoyu@cuhk.edu.cn>.

Presented at the 2^{nd} Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML) at the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. 2023. Copyright 2023 by the author(s).



(a) Output diversity

(b) Forward propagation

(c) Backward propagation

(d) Performance

Figure 1. Plots of (a) the output diversity metric, (b) the forward propagation metric, (c) the backward propagation metric, and (d) the training and test accuracy of the vanilla GCN with conventional weight initialization on Cora. (Sub-figures (a-c) present the results at initialization, while sub-figure (d) presents the results after training.) Besides the over-smoothing problem, we also observe significant gradient vanishing as depth increases under commonly-used initialization methods.

In this paper, we investigate the curse of depth from the perspective of weight initialization. As depicted in Figure 1, GCNs with conventional initialization encounter evident over-smoothing issues, examined by the decrease of the output diversity metric.[1] Moreover, we observe that the forward and backward propagation metrics exhibit steeper declines as the depth increases. This points to a severe gradient vanishing problem during the training process. These observations suggest that merely addressing over-smoothing is insufficient to tackle the curse of depth. Instead, it is necessary to have a more comprehensive initialization method that simultaneously addresses both over-smoothing and potential gradient pathology.

Fortunately, the signal propagation theory (Poole et al., 2016; Schoenholz et al., 2017; Pennington et al., 2017; 2018; Hanin, 2018) derived from DNNs provides valuable insights into the causes and potential solutions for gradient-related issues in GCNs. Applying this theory, we propose to assess the quality of signal propagation in GCNs from three perspectives: (1) **forward propagation**, (2) **backward propagation**, and (3) **output diversity propagation**. While forward propagation and backward propagation are derived from classical signal propagation theory, output diversity propagation is specifically introduced to address the over-smoothing problem in GCNs. With these three criteria in mind, we critically assess the commonly-used initialization strategies for GCNs and design new initialization methods. Our contributions are summarized below.

- We theoretically prove that conventional initialization schemes on GCNs often compromise on either forward propagation or output diversity propagation. Empirical evidence is provided to support our theory.
- We propose a random initialization searching algorithm, **S**ignal **P**ropagation **on** **G**raph (SPoGInit), designed to optimize the signal propagation metrics simultaneously. Validation on multiple datasets showcases the performance enhancement provided by SPoGInit in deep vanilla GCNs, indicating a strong correlation between our signal propagation metrics and GCN performance.
- We introduce ReZeroGCN, a GCN architecture equipped with skip connection and zero-initialized gating parameters. This architecture addresses all three signal propagation aspects. Experimental results on large-scale datasets show *consistent* performance improvements when the network depth increases from 4 to 64. For instance, on OGBN-Arxiv, ReZeroGCN achieves gains of 10% and 3% in training and test accuracy, respectively. To the best of our knowledge, this is the first result to fully resolve the curse of depth on OGBN-Arxiv over such a range of depths.

2. Related works

Over-smoothing in GCNs. The over-smoothing issue was first purposed in (Li et al., 2018) to explain the curse of depth in deep GCNs and then studied in (Oono & Suzuki, 2019; Cai & Wang, 2020; Yang et al., 2020; Chen et al., 2020a; Rusch et al., 2023b; Luan et al., 2020; Cong et al., 2021; Zhang et al., 2022). Although the smoothing effects of graph convolution may benefit shallow GCNs (Keriven, 2022; Wu et al., 2023), they adversely affect the performance of deep GCNs. To alleviate over-smoothing, a variety of techniques are adopted (Chen et al., 2022b). For vanilla GCNs, techniques such as nodes or edges dropping (Srivastava et al., 2014; Zou et al., 2019; Rong et al., 2020; Huang et al., 2020; Lu et al., 2021), normalization (Ioffe & Szegedy, 2015; Zhao & Akoglu, 2020; Zhou et al., 2020; Yang et al., 2020; Zhou et al., 2021a; Li et al., 2020; Guo et al., 2023), and regularization (Chen et al., 2020a; Yang et al., 2020; Zhou et al., 2021b) were explored. Efforts were also taken on different variants of GCN architectures, including GCNs with residual connections (Kipf & Welling, 2017; Jaiswal et al., 2022), GCNs with jumping connections (Xu et al., 2018; Liu et al., 2020; Zhu et al., 2020), and so on (Bose & Das, 2023; Di Giovanni et al., 2022; Chien et al., 2021; Gasteiger et al., 2019; Luan et al., 2019; Chen et al., 2020b; Li et al., 2019; Yan et al., 2022; Guo et al., 2022; Min et al., 2020; Chen et al., 2022a; Jin et al., 2022; Zheng et al., 2021; Yang et al., 2023b; Li et al., 2021). In contrast to these

[1] The output diversity, forward, and backward propagation metrics will be formally introduced in Section 4.

existing works, our paper delves into the impact of weight initialization to tackle over-smoothing (as well as gradient pathology) in GCNs.

Signal propagation. Classical signal propagation theory has built up a foundation for understanding how information flows through deep neural networks (DNNs) and guides the random weight initialization. At first, (Glorot & Bengio, 2010; He et al., 2015) studied the forward-backward propagation in linear or ReLU-activated models. Then, the mean-field theory (Neal, 1996; Lee et al., 2018; Matthews et al., 2018) was incorporated to study the signal propagation in models with general non-linear activation. Theoretical analysis on fully-connected neural networks (FCNNs) includes the study of Edge-of-Chaos (EOCs) (Poole et al., 2016; Schoenholz et al., 2017; Hayou et al., 2019; 2022) and dynamical isometry (Saxe et al., 2014; Pennington et al., 2017; 2018). Other works studied the signal propagation in deep CNN (Xiao et al., 2018), RNN (Chen et al., 2018), ResNet (Yang & Schoenholz, 2017; Hayou et al., 2022), autoencoder (Li & Nguyen, 2019), and LSTM/GRU (Gilboa et al., 2019). In the realm of GCNs, (Guo et al., 2022; Jaiswal et al., 2022) designed weight initialization methods via traditional forward and backward propagation. Our work further analyzes the output diversity propagation. Output diversity propagation is specifically tailored for GCN-like architectures, and is shown to be crucial to resolving the curse of depth.

Weight searching and gating parameters. In addition to signal propagation, other factors that reflect the training dynamics have also been exploited to guide the searching of initial weights (Dauphin & Schoenholz, 2019; Zhu et al., 2021). Our SPoGInit draws inspiration from MetaInit (Dauphin & Schoenholz, 2019) and is further tailored to vanilla GCNs. For DNNs with residual connections, (De & Smith, 2020; Zhang et al., 2019; Bachlechner et al., 2021) introduced trainable gating parameters to preserve signal propagation. We borrow the idea from ReZero (Bachlechner et al., 2021) and propose ReZeroGCN, which incorporates skip connections and gating parameters in GCNs.

Other works. Some existing works studied graph neural tangent kernel (GNTK) (Bayer et al., 2022; Du et al., 2019; Huang et al., 2022; Jiang et al., 2022; Sabanayagam et al., 2021; 2022; Zhou & Wang, 2022; Xu et al., 2021; Gebhart, 2022; Krishnagopal & Ruiz, 2023; Yang et al., 2023a). They analyzed the training dynamics of GCNs under the infinite-width limit.

3. Preliminaries and background

3.1. Notation

For any integer $n \in \mathbb{N}$, we define $[n] \triangleq \{1, 2, \dots, n\}$. We may denote a matrix $X \in \mathbb{R}^{m\times n}$ by $(x_{ij})_{i\in[m],j\in[n]}$, where x_{ij} is the entry in the i-the row and the j-th column. We further use $X_{i,:} \in \mathbb{R}^{1\times n}$ and $X_{:,j} \in \mathbb{R}^{m\times 1}$ to denote the i-th row and the j-th column of X, respectively. $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm. Given any function $f : \mathbb{R}^{m\times n} \to \mathbb{R}$, its derivative $\partial f/\partial X$ with respect to $X \in \mathbb{R}^{m\times n}$ is the $m \times n$ matrix with $(\partial f/\partial X)_{ij} = \partial f(X)/\partial x_{ij}$. For any activation function $\sigma : \mathbb{R} \to \mathbb{R}$, we use $\sigma(X) \in \mathbb{R}^{m\times n}$ to denote the output of applying σ entry-wise to the matrix X, i.e., $(\sigma(X))_{ij} = \sigma(x_{ij})$. We denote ReLU activation by $\mathrm{ReLU}(x) \triangleq \max(0, x)$ and tanh activation by $\tanh(x) \triangleq (e^x - e^{-x})/(e^x + e^{-x})$. For brevity, we use θ to denote the collection of all trainable parameters in a GCN model.

3.2. Graph convolutional networks

Featured graph. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph, where $\mathcal{V}$ is the set of nodes with $|\mathcal{V}| = n$, and $\mathcal{E}$ is the collection of edges. Assume that each node is associated with a d_0-dimensional feature and a label belonging to $[C]$. Let $x_i \in \mathbb{R}^{d_0\times 1}$ and $y_i \in [C]$ denote the feature and the label of node i, respectively. Define the node feature matrix as $X = (x_1^T, x_2^T, \dots, x_n^T)^T \in \mathbb{R}^{n\times d_0}$. Let $A = (\mathbb{1}_{\{(i,j)\in\mathcal{E}\}})_{i,j\in[n]} \in \mathbb{R}^{n\times n}$ represent the adjacency matrix and $D = \mathrm{diag}(A\mathbf{1}_n) \in \mathbb{R}^{n\times n}$ represent the degree matrix. Further, $\tilde{A} = A + I$ and $\tilde{D} = D + I$ denote the adjacency matrix degree matrix of graph $\mathcal{G}$ with self-loop added to each node. Finally, the normalized adjacency matrix is given by $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$.

Vanilla GCN. Vanilla GCN (Kipf & Welling, 2017) stacks neighborhood aggregations and feature transformations alternately. Specifically, let $H^{(l)}, X^{(l)} \in \mathbb{R}^{n\times d_l}$ denote the pre-activation and the post-activation embedding matrix at the l-th layer of the vanilla GCN, respectively. They are defined recursively by

$$H^{(l)} := \hat{A}X^{(l-1)}W^{(l)} + \mathbf{1}_n \cdot b^{(l)}, \quad X^{(l)} := \sigma(H^{(l)}),$$

where $W^{(l)} \in \mathbb{R}^{d_{l-1}\times d_l}$ and $b^{(l)} \in \mathbb{R}^{1\times d_l}$ are the weight and the bias term at the l-th layer, respectively. The input to the first layer is given by $X^{(0)} = X$, and the output matrix of an L-layer vanilla GCN is $H^{(L)} \in \mathbb{R}^{n\times C}$, which is then fed into a softmax layer to obtain the predicted labels.

ResGCN. Inspired by He et al. (2016), ResGCN (Kipf & Welling, 2017) combines residual connections with vanilla GCN. An L-layer ResGCN adds skip connections to the post-activation embeddings, i.e.,

$$\begin{aligned} H^{(l)} &:= \hat{A}X^{(l-1)}W^{(l)} + \mathbf{1}_n \cdot b^{(l)}, \\ X^{(l)} &:= \alpha\sigma(H^{(l)}) + \beta X^{(l-1)}, \quad \forall l \in [L], \end{aligned}$$

where $\alpha, \beta \in \mathbb{R}$ are trainable gating hyperparameters. [2] Linear transformations (trainable) are applied before $X^{(0)}$

[2] This is slightly different from the original version of ResGCN (Kipf & Welling, 2017), which does not present the gate parameters. Equivalently, the original ResGCN imposes $(\alpha, \beta) = (1, 1)$.

and after $H^{(L)}$ to ensure consistency of embedding sizes. We denote the output matrix of an L-layer ResGCN by $H^{(\text{out},L)} \in \mathbb{R}^{n\times C}$.

3.3. Initialization

Kaiming initialization (He et al., 2015) and Xavier initialization (Glorot & Bengio, 2010) are two popular random initialization methods in DNNs. Kaiming initialization, often in conjunction with ReLU activation, samples $W_{ij}^{(l)}$ independently from a random distribution with mean 0 and variance $2/d_{l-1}$. Xavier initialization samples $W_{ij}^{(l)}$ independently from a random distribution with mean 0 and variance $2/(d_{l-1}+d_l)$.

In practice, the weight distributions are usually set to be uniform distributions in DNNs, e.g., $\text{Uniform}\left(-\sqrt{\frac{6}{d_{l-1}}}, \sqrt{\frac{6}{d_{l-1}}}\right)$ for Kaiming initialization and $\text{Uniform}\left(-\sqrt{\frac{6}{d_{l-1}+d_l}}, \sqrt{\frac{6}{d_{l-1}+d_l}}\right)$ for Xavier initialization. In GCN models, uniform weight distributions are also widely used. Particularly, most cited works such as PairNorm (Zhao & Akoglu, 2020), DropEdge (Rong et al., 2020), DropNode (Huang et al., 2020), SkipNode (Lu et al., 2021), GCNII (Chen et al., 2020b), set the weight initialization to be $\text{Uniform}(-1/\sqrt{d_{l-1}}, 1/\sqrt{d_{l-1}})$. We simply refer to this initialization as "Conventional initialization" in the rest of this paper.

For theory, existing studies on signal propagation (Poole et al., 2016; Schoenholz et al., 2017) and neural tangent kernel (NTK) (Jacot et al., 2018) adopt Gaussian distribution for the simplicity of theoretical analysis. Following this convention, we adopt the following assumption for our theoretical derivation.

Assumption 3.1. At any layer $l \in [L]$, each $W_{k'k}^{(l)}$ is drawn i.i.d. from Gaussian distribution $N(0, \sigma_w^2/d_{l-1})$, and each bias term $b_k^{(l)}$ is set to be 0 at initialization.

Here, σ_w is a hyper-parameter controlling the variance of the random distribution. In particular, we note that $\sigma_w^2 = 2$ corresponds to Kaiming initialization, $\sigma_w^2 = 1$ corresponds to Xavier initialization for identical hidden layer width (i.e., $d_{l-1} = d_l$), and $\sigma_w^2 = 1/3$ corresponds to Conventional initialization.

Throughout this paper, we use uniformly distributed initialization in all our *experiments*, and use Gaussian distributed initialization (Assumption 3.1) in all our *theoretical analysis*.

4. Evaluation of commonly-used initialization in GCNs

In this section, we evaluate the quality of commonly-used initialization in GCNs. The evaluation is based on the signal propagation quality at initialization from the following three aspects.

Forward signal propagation is responsible to extract abstract and higher-level representations from the input data as the information flows through the network. We take the expected output-input norm ratio $\mathbb{E}_\theta[\|H^{(L)}(\theta)\|_{\text{F}}^2/\|X\|_{\text{F}}^2]$ at initialization as the *forward propagation metric*. A proper initialization method should prevent this metric from either vanishing or exploding as $L \to \infty$.

Backward signal propagation is responsible for updating the weights by utilizing gradients computed via the backpropagation algorithm. In vanilla GCN, the gradient of the weight $W^{(l)}$ at the l-th layer can be decomposed as $\partial\ell/\partial W^{(l)} = \sigma(H^{(l-1)})^T \cdot \hat{A} \cdot [\partial\ell/\partial H^{(l)}]$ where ℓ is the training loss. A stable magnitude of $\partial\ell/\partial H^{(l)}$ with respect to the layer l suggests that the gradient is less susceptible to vanishing or exploding. We take $\mathbb{E}_\theta[\|\partial\ell/\partial W^{(1)}\|_{\text{F}}^2]$ as the *backward propagation metric* at initialization. A proper initialization method should prevent this metric from vanishing or exploding as $L \to \infty$.

Output diversity propagation is responsible for tackling the over-smoothing issue, a GCN-specific problem. A number of existing works measure over-smoothing by Dirichlet energy (Cai & Wang, 2020; Zhou et al., 2021b; Rusch et al., 2023a), defined as $\text{Dir}(H^{(L)}) = \text{tr}(H^{(L)T}\hat{L}H^{(L)}) = \sum_{(i,j)\in\mathcal{E}} \|H_{i,:}^{(L)}/\sqrt{1+D_{ii}} - H_{j,:}^{(L)}/\sqrt{1+D_{jj}}\|^2$, where $\hat{L} = I - \hat{A}$ is the normalized Laplacian of graph $\mathcal{G}$. To mitigate the influence of the weight randomness and the embedding norm, we select the expected normalized Dirichlet energy $\mathbb{E}_\theta[\text{Dir}(H^{(L)})/\|H^{(L)}\|_{\text{F}}^2]$ at initialization as the *output diversity metric*. A proper initialization method should prevent this metric from vanishing as $L \to \infty$.

4.1. Vanilla GCN

As mentioned in Section 3.3, most GCN models use Conventional weight initialization. Now we theoretically evaluate the signal propagation quality of Conventional initialization in vanilla GCN. Analogous to the mean-field analysis in DNNs, we consider the infinite-width limit of vanilla GCN. Under this approximation, all the channels $\{H_{:,k}^{(l)}\}_{k=1}^{d_l}$ of each embedding at the l-th layer are i.i.d., following Gaussian distribution $N(\mathbf{0}, \Sigma^{(l)})$ (see Appendix C.1 for the details), which is also referred to as the neural network Gaussian process (NNGP) correspondence.

Under the NNGP correspondence, the forward propagation metric can be approximated by

$$\mathbb{E}_\theta\left[\|H^{(L)}\|_{\text{F}}^2/\|X\|_{\text{F}}^2\right] \approx \mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}\left[\|H\|_{\text{F}}^2/\|X\|_{\text{F}}^2\right],$$

and the output diversity metric can be approximated by

$$\mathbb{E}_\theta\left[\frac{\text{Dir}(H^{(L)})}{\|H^{(L)}\|_{\text{F}}^2}\right] \approx \mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}\left[\frac{\text{Dir}(H)}{\|H\|_{\text{F}}^2}\right],$$

where $H \sim N(\mathbf{0}_n, \Sigma^{(L)})$ means that the columns $\{H_{:,k}\}_{k=1}^{C}$ of $H \in \mathbb{R}^{n\times C}$ are i.i.d. $N(\mathbf{0}_n, \Sigma^{(L)})$.

Now we analyze the signal propagation of GCN under various activation functions. We start with ReLU since it is the most commonly used activation in popular GCN models (e.g., (Zhao & Akoglu, 2020; Rong et al., 2020; Huang et al., 2020; Lu et al., 2021; Chen et al., 2020b)). The following theorem states that under ReLU activation, if the initial weight variance $\sigma_w^2 \le 2$, which covers Conventional, Kaiming, and Xavier initialization, deep vanilla GCNs suffer from poor forward and output diversity propagation.

Theorem 4.1. *Under Assumption 3.1 and the NNGP correspondence approximation, when the activation function σ is ReLU, we have*

1. *The output diversity metric*

$$\mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}[\mathrm{Dir}(H)/\|H\|_{\mathrm{F}}^2]$$

is independent of the choice of σ_w^2.

2. *If $\sigma_w^2 = 2$, either the limit output diversity metric*

$$\lim_{L\to\infty} \mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}\left[\mathrm{Dir}(H)/\|H\|_{\mathrm{F}}^2\right] = 0,$$

or the limit forward propagation metric

$$\lim_{L\to\infty} \mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}[\|H\|_{\mathrm{F}}^2/\|X\|_{\mathrm{F}}^2] = 0.$$

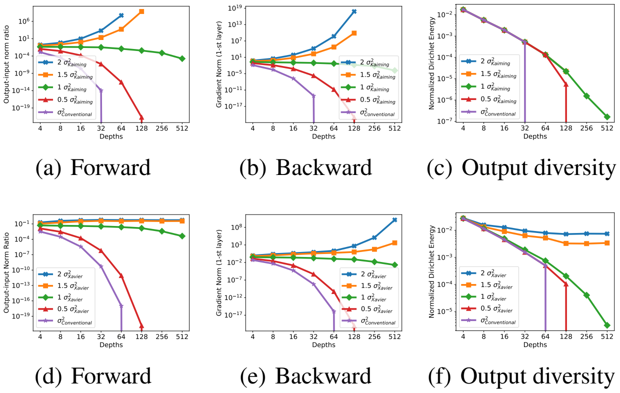
3. *When $\sigma_w^2 < 2$, we have*

$$\mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}[\|H\|_{\mathrm{F}}^2/\|X\|_{\mathrm{F}}^2] \le \frac{2C}{d_0}\cdot(\sigma_w^2/2)^L,$$

for any $L \ge 1$.

Part 1 of Theorem 4.1 states that it is impossible to improve output diversity by simply refining σ_w^2. Part 2 shows that under Kaiming initialization in ReLU-activated vanilla GCN, either the forward propagation metric or the output diversity metric vanishes as $L \to \infty$. Part 3 characterizes the shrinkage of the forward propagation metric when σ_w^2 is less than that of Kaiming initialization.

The purple lines in Figure 2(a)-(c) illustrate the shrinkage of the three signal propagation metrics under Conventional initialization as the network depth increases. Figure 2(a) presents the vanishing pattern of the forward propagation metric when σ_w^2 is no greater than that of Kaiming initialization, which validates Part 2 and 3 in Theorem 4.1. Figure 2(b) shows that the backward propagation metric transits from vanishing to stable, and then to exploding as σ_w^2 increases. Figure 2(c) shows that the output diversity propagation cannot be improved via merely changing σ_w^2, which validates Part 1 of Theorem 4.1.[3]



(a) Forward (b) Backward (c) Output diversity

(d) Forward (e) Backward (f) Output diversity

Figure 2. Plots of (a,d) forward metrics, (b,e) backward metrics, and (c,f) output diversity metrics of deep vanilla GCNs with different initialization variances and activations on Cora. (Sub-figures (a)-(c) are for ReLU activation, while sub-figures (d)-(f) are for tanh activation.) The choice of initialization variance plays a crucial role in forward and backward propagation. The output diversity propagation can be made stable with proper initialization variance for tanh activation, but not for ReLU activation.

The following theorem states that under tanh activation, if the initial weight variance $\sigma_w^2 \le 1$, which covers Conventional and Xavier initialization, deep vanilla GCN suffer fro poor forward propagation.

Theorem 4.2. *Under Assumption 3.1 and the NNGP correspondence approximation, when the activation function σ is tanh, we have*

1. *When $\sigma_w^2 = 1$, we have*

$$\lim_{L\to\infty} \mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}[\|H\|_{\mathrm{F}}^2/\|X\|_{\mathrm{F}}^2] = 0.$$

2. *When $\sigma_w^2 < 1$, we have*

$$\mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}[\|H\|_{\mathrm{F}}^2/\|X\|_{\mathrm{F}}^2] \le \frac{C}{d_0}\cdot\sigma_w^{2L},$$

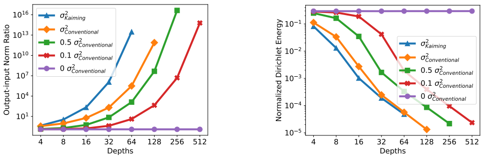
for any $L \ge 1$.

Different from ReLU-activated GCNs, Figure 2(f) shows propagation transits from vanishing to stable for tanh-activated models as σ_w^2 increases. With proper σ_w^2, stable propagation for all three types of signals can be achieved; see the orange lines in Figure 2(d)-2(f).

[3] In all the figures illustrating signal propagation metrics, disappearing nodes and vertical lines are caused by surpassing the machine precision. Specifically, the vanishing forward propagation metric result in vertical lines in the plots of the output diversity metric, while the exploding forward propagation metric leads to node disappearance in the plots of the output diversity metric.

4.2. GCNs with residual connections

Similarly to vanilla GCN, the curse of depth has also been reported in deep ResGCN (Huang et al., 2020; Rusch et al., 2023a). In this subsection, we focus on the signal propagation in ResGCN.



(a) Forward (b) Output diversity

Figure 3. (a) The forward metrics and (b) the output diversity metrics of ReLU-activated deep ResGCN on Cora. ResGCNs with non-zero initialization variances always suffer from exploding forward propagation and over-smoothing.

In the theoretical analysis, we assume identity activation function for simplicity (and hence ResGCN reduces to a linear model), but use ReLU activation in the experiments. As in the analysis of vanilla GCN, all the channels of $X^{(l)}$ and $H^{(\text{out},L)}$ are i.i.d. Gaussian under the infinite-width limit (see Proposition D.1), whose distributions are denoted by $N(\mathbf{0}_n, \tilde{\Sigma}^{(l)})$ and $N(\mathbf{0}_n, \tilde{\Sigma}^{(\text{out},L)})$, respectively. Thus, linear ResGCN also has its NNGP correspondence. The following theorem implies that the linear version of the original ResGCN in (Kipf & Welling, 2017) suffers from forward signal explosion and output diversity shrinkage under the NNGP approximation at initialization.

Theorem 4.3. *Suppose that there exists an eigenvector u of $\hat{A}$ corresponding to the eigenvalue 1, such that the input feature $X \in \mathbb{R}^{n\times d_0}$ satisfies $X^T u \neq \mathbf{0}_{d_0\times 1}$. Under the initialization in Assumption 3.1 and the NNGP correspondence approximation for linear ResGCN, if $\alpha^2\sigma_w^2 + \beta^2 > 1$ and $\alpha \neq 0$, then we have*

1. $\lim_{L\to\infty} \mathbb{E}_{H\sim N(\mathbf{0}_n, \tilde{\Sigma}^{(\text{out},L)})}[\|H\|_{\text{F}}^2/\|X\|_{\text{F}}^2] = +\infty$;
2. $\lim_{L\to\infty} \mathbb{E}_{H\sim N(\mathbf{0}_n, \tilde{\Sigma}^{(\text{out},L)})}[\text{Dir}(H)/\|H\|_{\text{F}}^2] = 0$.

Since $(\alpha, \beta) = (1, 1)$ for the original ResGCN, $\alpha^2\sigma_w^2 + \beta^2 > 1$ and $\alpha \neq 0$ always hold for any *nonzero* initialization variance. Part 1 and Part 2 of Theorem 4.3 indicate exploding forward propagation and over-smoothing, respectively, in the original ResGCN.

We now provide numerical evidence for Theorem 4.3. In Figure 3, we plot the forward and output diversity propagation of ReLU-activated ResGCN with different initialization variances. We see that the widely used Conventional and Kaiming initialization schemes (Huang et al., 2020; Kipf & Welling, 2017) (and essentially any non-zero initialization variance) lead to exploding forward propagation and over-smoothing.

In summary, the discussions in Section 4.1 and 4.2 highlight that conventional initialization schemes used in vanilla GCN and ResGCN all fail to achieve proper signal propagation. To tackle this challenge, we propose new initialization schemes in the next section.

5. New initialization schemes

5.1. SPoGInit: initialization guided by signal propagation on graph

Borrowing the idea from MetaInit (Dauphin & Schoenholz, 2019), we propose a random weight-searching algorithm called '**S**ignal **P**ropagation **o**n **G**raph' guided **Init**ialization (*SPoGInit*). Given any Xavier-initialized $\{\hat{W}^{(l)}\}_{l=1}^L$, we scale the weights layer-wise by $\gamma = (\gamma^{(l)})_{l\in[L]} \in \mathbb{R}^L_{>0}$ to yield new initialization $\theta(\gamma) = \{W^{(l)}\}_{l=1}^L = \{\gamma^{(l)}\hat{W}^{(l)}\}_{l=1}^L$ that achieves proper signal propagation. To achieve this goal, SPoGInit aims to find the optimal scaling vector γ by solving the following problem:

$$\min_{\gamma} F(\theta(\gamma)) := w_1 \underbrace{\left[\frac{\|H^{(1)}(\theta(\gamma))\|_{\text{F}}}{\|H^{(L-1)}(\theta(\gamma))\|_{\text{F}}} - 1\right]^2}_{(a)}$$
$$+ w_2 \underbrace{\left[\frac{\|g^{(2)}(\theta(\gamma))\|_{\text{F}}}{\|g^{(L-1)}(\theta(\gamma))\|_{\text{F}}} - 1\right]^2}_{(b)} - w_3 \underbrace{\frac{\text{Dir}(H^{(L)}(\theta(\gamma))}{\|H^{(L)}(\theta(\gamma)\|_{\text{F}}^2}}_{(c)},$$

where $g^{(l)}(\theta(\gamma)) \triangleq \partial\ell/\partial W^{(l)}$, and $w_1, w_2, w_3 > 0$ are pre-determined hyper-parameters. Term (a) and (b) are designed to stabilize forward and backward propagation, respectively, while term (c) is to suppress over-smoothing. For term (a), we use $\|H^{(1)}(\theta(\gamma))\|_{\text{F}}/\|H^{(L-1)}(\theta(\gamma))\|_{\text{F}}$ instead of $\|H^{(L)}(\theta(\gamma))\|_{\text{F}}/\|X\|_{\text{F}}$, because $H^{(1)}(\theta(\gamma))$, $H^{(L-1)}(\theta(\gamma))$ are both intermediate embeddings during the signal propagation process and share identical dimensions. For the same reason, we use $\|g^{(2)}(\theta(\gamma))\|_{\text{F}}/\|g^{(L-1)}(\theta(\gamma))\|_{\text{F}}$ in term (b). More details about SpoGInit are in Appendix F.

5.2. ReZeroGCN

Inspired by ReZero (Bachlechner et al., 2021), we propose a modified ResGCN architecture called ReZeroGCN by replacing the hyperparameters (α, β) with trainable parameters $(\alpha^{(l)}, \beta^{(l)})$ initialized to be $(0, 1)$. That is,

$$H^{(l)} = \hat{A}X^{(l-1)}W^{(l)} + \mathbf{1}_n \cdot b^{(l)},$$
$$X^{(l)} = \alpha^{(l)}\sigma(H^{(l)}) + \beta^{(l)}X^{(l-1)}.$$

Additionally, we use Xavier initialization to randomly initialize the weights $\{W^{(l)}\}$.

Since $X^{(L)} = X^{(L-1)} = \cdots = X^{(0)}$ under this initialization, deep ReZeroGCN naturally exhibits stable forward signal propagation and reasonable output diversity at initialization.

Next, we analyze the backward propagation of deep ReZeroGCN. The gradient of weight matrix $W^{(l)}$ at the l-th layer is given by

$$\begin{aligned}\frac{\partial \ell}{\partial W^{(l)}_{:,k}} &= \frac{\partial H^{(l)}_{:,k}}{\partial W^{(l)}_{:,k}} \frac{\partial X^{(l)}_{:,k}}{\partial H^{(l)}_{:,k}} \frac{\partial \ell}{\partial X^{(l)}_{:,k}} \\ &= \alpha^{(l)} \cdot \left[X^{(l-1)T} \hat{A} \cdot \operatorname{diag}\{\sigma'(H^{(l)})\} \frac{\partial \ell}{\partial X^{(l)}_{:,k}} \right]\end{aligned} \tag{1}$$

for any channel $k \in [d]$.

As shown in equation (1), the initial $(\alpha^{(l)}, \beta^{(l)}) = (0, 1)$ results in zero gradients of weight matrices at initialization. However, the gradient of $\alpha^{(l)}$ can be non-zero at initialization (See Appendix G.5) and help $W^{(l)}$ get updated in the following training epochs. In the next section, we will show by experiments that the gradient norms quickly improve in the early stage of training.

We summarize the behaviors on signal propagation for popular skip-connection-based GCN models in Table 1. Relevant experiments will be presented in the next section.

Table 1. Summary of signal propagation for popular skip-connection-based GCNs. ✓ means that the corresponding signal propagation is well-behaved. The proposed ReZeroGCN addresses all three signal propagation aspects properly.

Models	Forward	Backward	Output diversity
JKNet	vanish	vanish	✓
ResGCN	explode	explode	vanish
GCNII	✓	vanish	✓
ReZeroGCN	✓	✓	✓

6. Experiments

Due to the limited space, we introduce all descriptions of datasets, the experimental settings, and hyperparameters in Appendix G.1 and G.2.

6.1. Vanilla GCNs and proposed SPoGInit methods

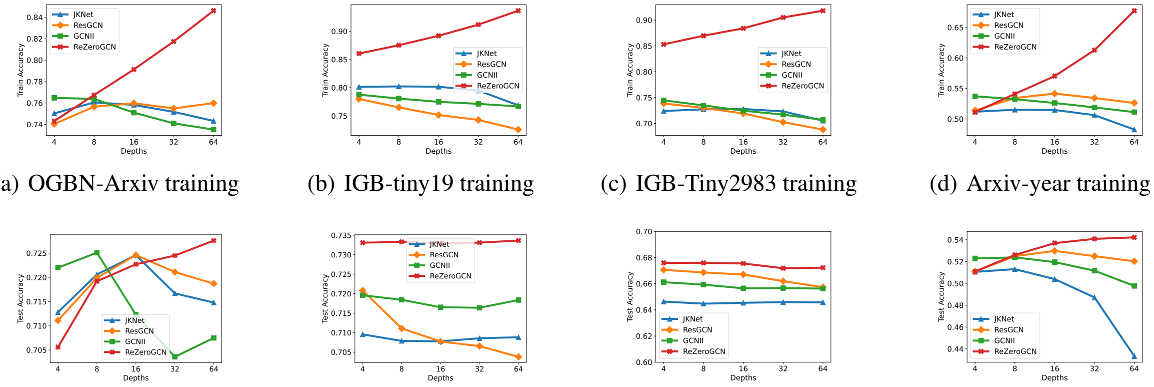
We first examine **whether SPoGInit tackles the signal propagation and performance degradation of deep GCNs**. We set Conventional and Xavier initialization schemes as our baselines.

In Figure 5(a)-5(c), we report the average signal propagation metrics for vanilla GCNs with different initializations and varying depths. The results indicate that SPoGInit stabilizes the forward-backward propagations and enhances the output diversity. Notably, SPoGInit successfully prevents gradient vanishing, a common issue encountered by other initialization. As a result, SPoGInit effectively alleviates the performance degradation of deep vanilla GCNs. It outperforms the baselines (Xavier, Conventional) by 7.5% and 35.2% test accuracy at depth 128 (see Figure 5(d)). Similar phenomena are also observed in various other datasets. We present more experiments in Appendix G.3. These results also demonstrate a strong correlation between the proposed signal propagation metrics and the actual performance of deep GCNs.

6.2. Skip-connection-based GCN models and ReZeroGCN

We examine **whether deep ReZeroGCN overcomes the curse of depth**. We adopt three popular skip-connection-based GCN models, JKNet, ResGCN, and GCNII, as baselines. Due to the powerful expressivity brought by skip connections (Chen et al., 2020b), baseline models achieve perfect training accuracies on small-sized datasets. Thus, we only consider large-scale datasets to examine the curse of depth.

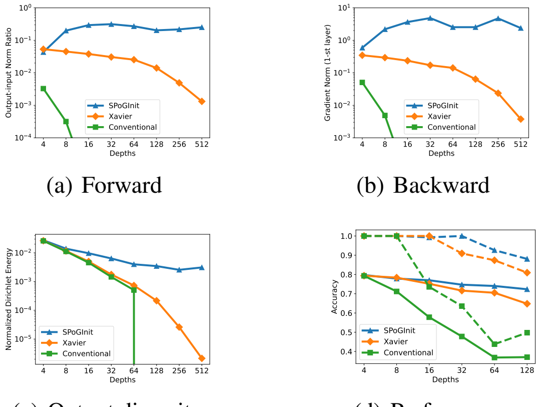
Figure 4 presents the average training and test accuracies of ReLU-activated models with various depths. (The results



(a) OGBN-Arxiv training (b) IGB-tiny19 training (c) IGB-Tiny2983 training (d) Arxiv-year training

(e) OGBN-Arxiv test (f) IGB-tiny19 test (g) IGB-Tiny2983 test (h) Arxiv-year test

Figure 4. The average training accuracies (a)-(d) and test accuracies (e)-(h) of different skip-connection-based GCNs with ReLU activation on various datasets. ReZeroGCN outperforms baselines on all datasets and achieves consistent training gains with increasing depth.



(a) Forward

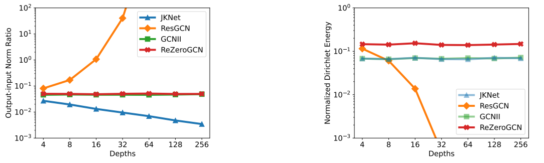
(b) Backward

(c) Output diversity

(d) Performance

Figure 5. (a) The forward metrics, (b) backward metrics, and (c) output diversity metrics of deep GCNs with different initializations on the Cora dataset. (d) Training accuracies (dashed lines) and test accuracies (solid lines) of deep GCNs after training on Cora. We find that SPoGInit simultaneously addresses three signal propagation aspects, and alleviates the performance degradation.

of tanh-activated models are presented in Appendix G.4.) We see that ReZeroGCN stands out by achieving *consistent* performance gains as the depth increases. Additionally, on the OGBN-Arxiv and Arxiv-year datasets, ReZeroGCN achieves around 3% gain in test accuracy by deepening the model from 4 to 64 layers. These results demonstrate that ReZeroGCN successfully overcomes the curse of depth.



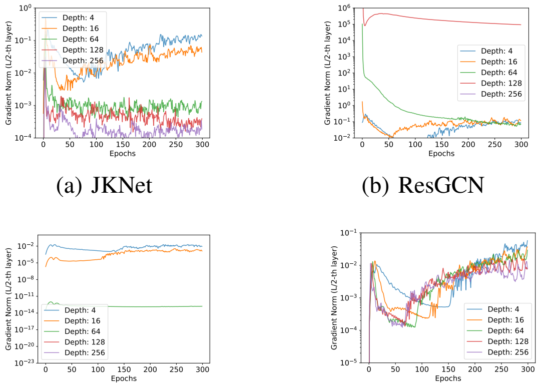
(a) Forward

(b) Output diversity

Figure 6. (a) The forward metrics and (b) the output diversity metrics of different models and depths on the Cora dataset. ResGCN suffers from forward exploding and output diversity vanishing. In contrast, ReZeroGCN addresses the forward propagation and preserves the output diversity.

Next, we investigate **whether ReZeroGCN achieves well-behaved signal propagation**. Figure 6 presents the average forward metric and output diversity metric of different models with various depths. Results indicate that ReZeroGCN effectively addresses forward propagation and output diversity.

Skip connections significantly change the back-propagation computation. Therefore, we select the middle hidden layer (the $L/2$-th layer in an L-layer model) as the representative layer to measure the backward propagation. Figure 7 plots the average backward metrics of the skip-connection-based GCNs with various depths L during early training. We see that the baseline models suffer from poor backward propagation. Similar phenomena are also observed at initialization; see Appendix G.5. In contrast, the backward propagation of deep RezeroGCN with different depths rapidly tends to stabilize during early training. The results of other layers in the skip-connection-based models are presented in Appendix G.5.



(a) JKNet

(b) ResGCN

(c) GCNII

(d) ReZeroGCN

Figure 7. The backward metrics at the $L/2$-th layer of different ReLU-activated skip-connection-based GCNs with various depths L during early training on the IGB-Tiny19 dataset. Baselines suffer from gradient vanishing or exploding problems. Across early 300 epochs, the gradient norms of (a) JKNet and (c) GCNII vanish as the depths increase, while the gradient norms of (b) ResGCN explode. In contrast, the gradient norms of (d) ReZeroGCN quickly improve in the early training. The disappearing lines in (b)-(c) are caused by surpassing the machine precision.

In conclusion, ReZeroGCN offers a straightforward yet effective solution to address the signal propagation challenges. As a result, deep ReZeroGCN possesses powerful training capability and ultimately overcomes the curse of depth.

7. Conclusion

We state that it is crucial to overcome performance degradation in deep GCNs by addressing the forward propagation, backward propagation, and output diversity propagation issues. However, theoretical analysis and empirical studies have revealed that widely used initialization methods in GCNs fail to meet these requirements, resulting in significant training issues (e.g., gradient vanishing and oversmoothing) in deep networks. To tackle these challenges, new initialization methods, SPoGInit and ReZeroGCN, are proposed for vanilla GCN and ResGCN, respectively. The experiments demonstrate that SPoGInit effectively alleviates performance degradation in deep GCNs. Furthermore, ReZeroGCN significantly overcomes the curse of depth. Interesting directions for future work include applying signal propagation on the GNNs with attention mechanisms.

Acknowledgements

We thank the anonymous reviewers for their comments and constructive suggestions. This work is supported by the National Key R&D Program of China under grant 2022YFA1003900; Hetao Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone Project (No.HZQSWS-KCCYB-2022046); University Development Fund UDF01001491 from The Chinese University of Hong Kong, Shenzhen; Guangdong Key Lab on the Mathematical Foundation of Artificial Intelligence, Department of Science and Technology of Guangdong Province.

References

Bachlechner, T., Majumder, B. P., Mao, H., Cottrell, G., and McAuley, J. ReZero is all you need: Fast convergence at large depth. In *Uncertainty in Artificial Intelligence*, pp. 1352–1361. PMLR, 2021.

Bayer, A., Chowdhury, A., and Segarra, S. Label propagation across graphs: Node classification using graph neural tangent kernels. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5483–5487. IEEE, 2022.

Bose, K. and Das, S. Can graph neural networks go deeper without over-smoothing? Yes, with a randomized path exploration! In *IEEE Transactions on Emerging Topics in Computational Intelligence*. IEEE, 2023.

Cai, C. and Wang, Y. A note on over-smoothing for graph neural networks. In *International Conference on Machine Learning*. PMLR, 2020.

Chen, D., Lin, Y., Li, W., Li, P., Zhou, J., and Sun, X. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 3438–3445, 2020a.

Chen, J., Wang, Y., Bodnar, C., Ying, R., Liò, P., and Wang, Y. G. Dirichlet energy enhancement of graph neural networks by framelet augmentation. 2022a.

Chen, M., Pennington, J., and Schoenholz, S. Dynamical isometry and a mean field theory of RNNs: Gating enables signal propagation in recurrent neural networks. In *International Conference on Machine Learning*, pp. 873–882. PMLR, 2018.

Chen, M., Wei, Z., Huang, Z., Ding, B., and Li, Y. Simple and deep graph convolutional networks. In *International Conference on Machine Learning*, pp. 1725–1735. PMLR, 2020b.

Chen, T., Zhou, K., Duan, K., Zheng, W., Wang, P., Hu, X., and Wang, Z. Bag of tricks for training deeper graph neural networks: A comprehensive benchmark study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022b.

Chien, E., Peng, J., Li, P., and Milenkovic, O. Adaptive universal generalized PageRank graph neural network. In *International Conference on Learning Representations*, 2021.

Cong, W., Ramezani, M., and Mahdavi, M. On provable benefits of depth in training graph convolutional networks. *Advances in Neural Information Processing Systems*, 34: 9936–9949, 2021.

Dauphin, Y. N. and Schoenholz, S. S. MetaInit: Initializing learning by learning to initialize. *Advances in Neural Information Processing Systems*, 32, 2019.

De, S. and Smith, S. Batch normalization biases residual blocks towards the identity function in deep networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 19964–19975, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, 2009.

Di Giovanni, F., Rowbottom, J., Chamberlain, B. P., Markovich, T., and Bronstein, M. M. Graph neural networks as gradient flows: Understanding graph convolutions via energy. *arXiv preprint arXiv:2206.10991*, 2022.

Du, S. S., Hou, K., Salakhutdinov, R. R., Poczos, B., Wang, R., and Xu, K. Graph neural tangent kernel: Fusing graph neural networks with graph kernels. *Advances in Neural Information Processing Systems*, 32, 2019.

Gasteiger, J., Bojchevski, A., and Günnemann, S. Predict then propagate: Graph neural networks meet personalized PageRank. In *International Conference on Learning Representations*, 2019.

Gebhart, T. Graph convolutional networks from the perspective of sheaves and the neural tangent kernel. In *Topological, Algebraic and Geometric Learning Workshops 2022*, pp. 124–132. PMLR, 2022.

Gilboa, D., Chang, B., Chen, M., Yang, G., Schoenholz, S. S., Chi, E. H., and Pennington, J. Dynamical isometry and a mean field theory of LSTMs and GRUs. *arXiv preprint arXiv:1901.08987*, 2019.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Guo, K., Zhou, K., Hu, X., Li, Y., Chang, Y., and Wang, X. Orthogonal graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 3996–4004, 2022.

Guo, X., Wang, Y., Du, T., and Wang, Y. ContraNorm: A contrastive learning perspective on oversmoothing and beyond. In *International Conference on Learning Representations*, 2023.

Hanin, B. Which neural net architectures give rise to exploding and vanishing gradients? *Advances in Neural Information Processing Systems*, 31, 2018.

Hardy, G. H., Littlewood, J. E., Pólya, G., Pólya, G., et al. *Inequalities*. Cambridge university press, 1952.

Hayou, S., Doucet, A., and Rousseau, J. On the impact of the activation function on deep neural networks training. In *International Conference on Machine Learning*, pp. 2672–2680. PMLR, 2019.

Hayou, S., Doucet, A., and Rousseau, J. The curse of depth in kernel regime. In *I (Still) Can't Believe It's Not Better! Workshop at NeurIPS 2021*, pp. 41–47. PMLR, 2022.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. *Advances in Neural Information Processing Systems*, 33:22118–22133, 2020.

Huang, W., Rong, Y., Xu, T., Sun, F., and Huang, J. Tackling over-smoothing for general graph convolutional networks. *arXiv preprint arXiv:2008.09864*, 2020.

Huang, W., Li, Y., Du, W., Yin, J., Da Xu, R. Y., Chen, L., and Zhang, M. Towards deepening graph neural networks: A GNTK-based optimization perspective. *International Conference on Learning Representations*, 2022.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pp. 448–456. PMLR, 2015.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

Jaiswal, A., Wang, P., Chen, T., Rousseau, J., Ding, Y., and Wang, Z. Old can be gold: Better gradient flow can make vanilla-GCNs great again. *Advances in Neural Information Processing Systems*, 35:7561–7574, 2022.

Jiang, S., Man, Y., Song, Z., Yu, Z., and Zhuo, D. Fast graph neural tangent kernel via Kronecker sketching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7033–7041, 2022.

Jin, W., Liu, X., Ma, Y., Aggarwal, C., and Tang, J. Towards feature overcorrelation in deeper graph neural networks. 2022.

Keriven, N. Not too little, not too much: a theoretical analysis of graph (over)smoothing. In *Advances in Neural Information Processing Systems*, 2022.

Khatua, A., Mailthody, V. S., Taleka, B., Ma, T., Song, X., and Hwu, W.-m. IGB: Addressing the gaps in labeling, features, heterogeneity, and size of public graph datasets for deep learning research. *arXiv preprint arXiv:2302.13522*, 2023.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations*, 2017.

Krishnagopal, S. and Ruiz, L. Graph neural tangent kernel: Convergence on large graphs. *arXiv preprint arXiv:2301.10808*, 2023.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25 (1106-1114):1, 2012.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., and Sohl-Dickstein, J. Deep neural networks as Gaussian processes. In *International Conference on Learning Representations*, 2018.

Li, G., Müller, M., Thabet, A., and Ghanem, B. DeepGCNs: Can GCNs go as deep as CNNs? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

Li, G., Xiong, C., Thabet, A., and Ghanem, B. DeeperGCN: All you need to train deeper GCNs, 2020.

Li, G., Müller, M., Ghanem, B., and Koltun, V. Training graph neural networks with 1000 layers. In *International Conference on Machine Learning*, 2021.

Li, P. and Nguyen, P.-M. On random deep weight-tied autoencoders: Exact asymptotic analysis, phase transitions, and implications to training. In *International Conference on Learning Representations*, 2019.

Li, Q., Han, Z., and Wu, X.-M. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Lim, D., Hohne, F., Li, X., Huang, S. L., Gupta, V., Bhalerao, O., and Lim, S. N. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. In *Advances in Neural Information Processing Systems*, volume 34, pp. 20887–20902, 2021.

Liu, M., Gao, H., and Ji, S. Towards deeper graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 338–348, 2020.

Lu, W., Zhan, Y., Guan, Z., Liu, L., Yu, B., Zhao, W., Yang, Y., and Tao, D. SkipNode: On alleviating over-smoothing for deep graph convolutional networks. *arXiv preprint arXiv:2112.11628*, 2021.

Luan, S., Zhao, M., Chang, X.-W., and Precup, D. Break the ceiling: Stronger multi-scale deep graph convolutional networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Luan, S., Zhao, M., Chang, X.-W., and Precup, D. Training matters: Unlocking potentials of deeper graph convolutional neural networks. *arXiv preprint arXiv:2008.08838*, 2020.

Matthews, A. G. d. G., Rowland, M., Hron, J., Turner, R. E., and Ghahramani, Z. Gaussian process behaviour in wide deep neural networks. In *International Conference on Learning Representations*, 2018.

Min, Y., Wenkel, F., and Wolf, G. Scattering GCN: Overcoming oversmoothness in graph convolutional networks. *Advances in Neural Information Processing Systems*, 33: 14498–14508, 2020.

Neal, R. M. *Bayesian learning for neural networks*, volume 118. Springer-Verlag, 1996.

Oono, K. and Suzuki, T. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*, 2019.

Pennington, J., Schoenholz, S., and Ganguli, S. Resurrecting the sigmoid in deep learning through dynamical isometry: Theory and practice. *Advances in Neural Information Processing Systems*, 30, 2017.

Pennington, J., Schoenholz, S., and Ganguli, S. The emergence of spectral universality in deep networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 1924–1932. PMLR, 2018.

Poole, B., Lahiri, S., Raghu, M., Sohl-Dickstein, J., and Ganguli, S. Exponential expressivity in deep neural networks through transient chaos. *Advances in Neural Information Processing Systems*, 29, 2016.

Rong, Y., Huang, W., Xu, T., and Huang, J. DropEdge: Towards deep graph convolutional networks on node classification. In *International Conference on Learning Representations*, 2020.

Rusch, T. K., Bronstein, M. M., and Mishra, S. A survey on oversmoothing in graph neural networks. *arXiv preprint arXiv:2303.10993*, 2023a.

Rusch, T. K., Chamberlain, B. P., Mahoney, M. W., Bronstein, M. M., and Mishra, S. Gradient gating for deep multi-rate learning on graphs. In *International Conference on Learning Representations*, 2023b.

Sabanayagam, M., Esser, P., and Ghoshdastidar, D. New insights into graph convolutional networks using neural tangent kernels. *arXiv preprint arXiv:2110.04060*, 2021.

Sabanayagam, M., Esser, P., and Ghoshdastidar, D. Representation power of graph convolutions: Neural tangent kernel analysis. *arXiv preprint arXiv:2210.09809*, 2022.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Represenatations*, 2014.

Schoenholz, S. S., Gilmer, J., Ganguli, S., and Sohl-Dickstein, J. Deep information propagation. In *International Conference on Learning Representations*, 2017.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. In *Journal of Machine Learning Research*, volume 15, pp. 1929–1958, 2014.

Wu, X., Chen, Z., Wang, W., and Jadbabaie, A. A non-asymptotic analysis of oversmoothing in graph neural networks. *International Conference on Learning Representations*, 2023.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. A comprehensive survey on graph neural networks. In *IEEE Transactions on Neural Networks and Learning Systems*, volume 32, pp. 4–24. IEEE, 2020.

Xiao, L., Bahri, Y., Sohl-Dickstein, J., Schoenholz, S., and Pennington, J. Dynamical isometry and a mean field theory of CNNs: How to train 10,000-layer vanilla convolutional neural networks. In *International Conference on Machine Learning*, pp. 5393–5402. PMLR, 2018.

Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.-i., and Jegelka, S. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning*, pp. 5453–5462. PMLR, 2018.

Xu, K., Zhang, M., Li, J., Du, S. S., Kawarabayashi, K.-i., and Jegelka, S. How neural networks extrapolate: From feedforward to graph neural networks. In *International Conference on Learning Representations*, 2021.

Yan, Y., Hashemi, M., Swersky, K., Yang, Y., and Koutra, D. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. In *2022 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2022.

Yang, C., Wang, R., Yao, S., Liu, S., and Abdelzaher, T. Revisiting over-smoothing in deep GCNs. *arXiv preprint arXiv:2003.13663*, 2020.

Yang, C., Wu, Q., Wang, J., and Yan, J. Graph neural networks are inherently good generalizers: Insights by bridging GNNs and MLPs. In *International Conference on Learning Representations*, 2023a.

Yang, G. and Schoenholz, S. S. Mean field residual networks: On the edge of chaos. *Advances in Neural Information Processing Systems*, 30, 2017.

Yang, R., Dai, W., Li, C., Zou, J., and Xiong, H. Tackling over-smoothing in graph convolutional networks with EM-based joint topology optimization and node classification. In *IEEE Transactions on Signal and Information Processing over Networks*, volume 9, pp. 123–139. IEEE, 2023b.

Yang, Z., Cohen, W., and Salakhudinov, R. Revisiting semi-supervised learning with graph embeddings. In *International Conference on Machine Learning*, pp. 40–48. PMLR, 2016.

Zhang, H., Dauphin, Y. N., and Ma, T. Fixup initialization: Residual learning without normalization. In *International Conference on Learning Representations*, 2019.

Zhang, W., Sheng, Z., Yin, Z., Jiang, Y., Xia, Y., Gao, J., Yang, Z., and Cui, B. Model degradation hinders deep graph neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2493–2503, 2022.

Zhao, L. and Akoglu, L. PairNorm: Tackling oversmoothing in GNNs. In *International Conference on Learning Representations*, 2020.

Zheng, L., Fu, D., and He, J. Tackling oversmoothing of gnns with contrastive learning. 2021.

Zhou, K., Huang, X., Li, Y., Zha, D., Chen, R., and Hu, X. Towards deeper graph neural networks with differentiable group normalization. In *Advances in Neural Information Processing Systems*, volume 33, pp. 4917–4928, 2020.

Zhou, K., Dong, Y., Wang, K., Lee, W. S., Hooi, B., Xu, H., and Feng, J. Understanding and resolving performance degradation in deep graph convolutional networks. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, pp. 2728–2737, 2021a.

Zhou, K., Huang, X., Zha, D., Chen, R., Li, L., Choi, S.-H., and Hu, X. Dirichlet energy constrained learning for deep graph neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 21834–21846, 2021b.

Zhou, X. and Wang, H. On the explainability of graph convolutional network with GCN tangent kernel. *Neural Computation*, 35(1):1–26, 2022.

Zhu, C., Ni, R., Xu, Z., Kong, K., Huang, W. R., and Goldstein, T. GradInit: Learning to initialize neural networks for stable and efficient training. *Advances in Neural Information Processing Systems*, 34:16410–16422, 2021.

Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., and Koutra, D. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems*, 33:7793–7804, 2020.

Zou, D., Hu, Z., Wang, Y., Jiang, S., Sun, Y., and Gu, Q. Layer-dependent importance sampling for training deep and large graph convolutional networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

A. Supplemental notation and preliminaries

A.1. Notation

For any matrix $X = (x_{ij}) \in \mathbb{R}^{m\times n}$, the vectorizaion of X is defined by

$$\mathrm{vec}(X) := (x_{11}, \dots, x_{m1}, x_{12}, \dots, x_{m2}, \dots, x_{1n}, \dots, x_{mn})^T \in \mathbb{R}^{mn\times 1}.$$

For any matrix $X = (x_{ij}) \in \mathbb{R}^{m\times n}$ and $Y = (y_{ij}) \in \mathbb{R}^{p\times q}$, the Kronecker product of X and Y is a $mp \times nq$ block matrix defined by

$$X \otimes Y := \begin{pmatrix} x_{11}Y & \dots & x_{1n}Y \\ \vdots & \ddots & \vdots \\ x_{m1}Y & \dots & x_{mn}Y \end{pmatrix}.$$

For a matrix $X = (x_{ij}) \in \mathbb{R}^{m\times n}$, if $x_{ij} = 0$ for all $i \in [m]$ and $j \in [n]$, we denote $X = \mathbf{0}_{m\times n}$; if $x_{ij} = 1$ for all $i \in [m]$ and $j \in [n]$, we denote $X = \mathbf{1}_{m\times n}$. For a vector $Z = (z_i) \in \mathbb{R}^n$, if $z_i = 0$ for all $i \in [n]$, we denote $Z = \mathbf{0}_n$; if $z_i = 1$ for all $i \in [n]$, we denote $Z = \mathbf{1}_n$.

A.2. Supplemental skip-connected-based GCN architectures

JKNet. Xu et al. (2018) proposes jumping knowledge network (JKNet) by only combining all embeddings in the hidden layers before getting the output. To be more specific, an L-layer JKNet is defined by

$$\begin{aligned} X^{(0)} &:= XW^{(0)} + \mathbf{1}_n \cdot b^{(0)}, \\ H^{(l)} &:= \hat{A}X^{(l-1)}W^{(l)} + \mathbf{1}_n \cdot b^{(l)}, \quad && \text{for any } l \in [L], \\ X^{(l)} &:= \sigma(H^{(l)}), && \text{for any } l \in [L], \\ H^{(\mathrm{out},L)} &:= \mathrm{COMBINE}(X^{(1)}, X^{(2)}, \dots, X^{(L)}). \end{aligned}$$

We assume that COMBINE is a linear transformation from the concatenation of $\{X^{(l)}\}_{l=1}^L$ to the output embedding.

GCNII. Chen et al. (2020b) designs GCNII by (1) using residual connection to the initial layer and (2) combining identity matrices with the weight matrices. Specifically, an L-layer GCNII is defined by

$$\begin{aligned} X^{(0)} &:= XW^{(0)} + \mathbf{1}_n \cdot b^{(0)}, \\ H^{(l)} &:= (1-\alpha_l)\hat{A}X^{(l-1)} \cdot \left[(1-\beta_l)I_d + \beta_l W_1^{(l)}\right] \\ &\quad + \alpha_l X^{(0)} \cdot \left[(1-\beta_l)I_d + \beta_l W_2^{(l)}\right], \quad && \text{for any } l \in [L], \\ X^{(l)} &:= \sigma(H^{(l)}), && \text{for any } l \in [L], \\ H^{(\mathrm{out},L)} &:= X^{(L)}W^{(L+1)} + \mathbf{1}_n \cdot b^{(L+1)}, \end{aligned} \tag{2}$$

where $\{\alpha_l, \beta_l\}_{l=1}^L$ are predetermined hyperparameters and β_l, set to be $\log(\frac{\lambda}{l+1} + 1)$, vanishes to 0 as $l \to \infty$, where λ is a predetermined hyperparameter. Chen et al. (2020b) call the architecture imposing $W_1^{(l)} = W_2^{(l)}$ by GCNII and call its improved version by (2) GCNII* in their paper. For the sake of brevity, we refer to the architecture (2) as GCNII without bringing any confusion.

B. Convolutional kernel

Suppose that graph $\mathcal{G}$ has M connected components. The m-th component is a subgraph denoted by $\mathcal{G}_m = (\mathcal{V}_m, \mathcal{E}_m)$ for $m \in [M]$. We present a well-known result characterizing the eigenvalues and the eigenvectors of $\hat{A}$ without giving proof, see, e.g., Proposition 1 in Oono & Suzuki (2019).

Proposition B.1. *Suppose that $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ has M connected components $\{\mathcal{G}_m = (\mathcal{V}_m, \mathcal{E}_m)\}_{m=1}^{M}$ and the eigenvalues of $\hat{A}$ are $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. Then we have*

- $\lambda_i = 1$, *for any $1 \leq i \leq M$.*
- $\lambda_i \in (-1, 1)$, *for any $M + 1 \leq i \leq n$.*

Moreover, the set $\{v^{(m)} = \tilde{D}^{\frac{1}{2}} u^{(m)} : m \in [M]\}$ is a basis of the m-dimensional eigenspace of $\hat{A}$ corresponding to the eigenvalue 1, where $u^{(m)} = (\mathbb{1}_{\{i \in \mathcal{V}_m\}})_{i \in [n]} \in \mathbb{R}^{n \times 1}$ is the indicator vector of the m-th connected component $\mathcal{G}_m$.

Lemma B.2. *Given any $H \in \mathbb{R}^{n \times C}$ and $H \neq \mathbf{0}_{n \times C}$, we have $0 \leq \mathrm{Dir}(H)/\|H\|_{\mathrm{F}}^2 \leq 2$.*

Proof. Recall that $\hat{L} = I - \hat{A}$ is the normalized Laplacian of graph $\mathcal{G}$. By Proposition B.1, all the eigenvalues of $\hat{L}$ belong to $[0, 2)$.

Given any $H \in \mathbb{R}^{n \times C}$, we have

$$\mathrm{Dir}(H) = \mathrm{tr}(H^T \hat{L} H) = \sum_{k=1}^{C} H_{:,k}^T \hat{L} H_{:,k} \leq \sum_{k=1}^{C} 2 \cdot H_{:,k}^T H_{:,k} = 2\|H\|_{\mathrm{F}}^2.$$

Similarly, we have

$$\mathrm{Dir}(H) = \mathrm{tr}(H^T \hat{L} H) = \sum_{k=1}^{C} H_{:,k}^T \hat{L} H_{:,k} \geq \sum_{k=1}^{C} 0 \cdot H_{:,k}^T H_{:,k} = 0.$$

Therefore, we conclude that

$$0 \leq \mathrm{Dir}(H)/\|H\|_{\mathrm{F}}^2 \leq 2.$$

□

C. Signal propagation theory for vanilla GCN

C.1. NNGP correspondence for vanilla GCN

Proposition C.1 (NNGP correspondence for vanilla GCN). *Under the initialization in Assumption 3.1, as the network widths $d_1, d_2, \dots, d_{L-1}$ sequentially go to infinity, the l-th layer's pre-activation embedding channels $\{H^{(l)}_{:,k}\}_{k\in[d_l]}$ converge to i.i.d. n-dimensional Gaussian random variables $N(\mathbf{0}_n, \Sigma^{(l)})$ in distribution. The covariance matrices are*

$$\begin{aligned}\Sigma^{(1)} &= \frac{\sigma_w^2}{d_0}\hat{A}XX^T\hat{A},\\ \Sigma^{(l+1)} &= \sigma_w^2\hat{A}G(\Sigma^{(l)})\hat{A},\end{aligned} \tag{3}$$

where $G(\Sigma) = \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\sigma(h)\sigma(h)^T]$ for any $n\times n$ positive semi-definite matrix Σ.

Proof of Proposition C.1. We will prove Proposition C.1 by mathematical induction as follows.

Base case. Since the bias terms are initialized to be zero in Assumption 3.1, when $l=1$, the k-th channel of the embedding is

$$H^{(1)}_{:,k} = \hat{A}XW^{(1)}_{:,k} + \mathbf{1}_n\cdot b^{(1)}_k = \hat{A}XW^{(1)}_{:,k}. \tag{4}$$

According to Assumption 3.1, $\{W^{(1)}_{:,k}\}_{k\in[d_1]}$ are i.i.d. Gaussian distributed, so $\{H^{(1)}_{:,k}\}_{k\in[d_1]}$ are also i.i.d. Gaussian distributed. Taking the expectation of (4), we get

$$\mathbb{E}[H^{(1)}_{:,k}] = \hat{A}X\cdot\mathbb{E}[W^{(1)}_{:,k}] = 0.$$

Calculating the covariance matrix of (4), we have

$$\begin{aligned}\mathrm{Cov}[H^{(1)}_{:,k}, H^{(1)}_{:,k}] &= \mathbb{E}[H^{(1)}_{:,k}\cdot H^{(1)T}_{:,k}] = \mathbb{E}[\hat{A}XW^{(1)}_{:,k}W^{(1)T}_{:,k}X^T\hat{A}]\\ &= \hat{A}X\cdot\mathbb{E}[W^{(1)}_{:,k}W^{(1)T}_{:,k}]\cdot X^T\hat{A} = \hat{A}X\cdot\left(\frac{\sigma_w^2}{d_0}\cdot I_{d_0}\right)\cdot X^T\hat{A}\\ &= \frac{\sigma_w^2}{d_0}\hat{A}XX^T\hat{A}\end{aligned}$$

Thus, if we define $\Sigma^{(1)} = \sigma_w^2\hat{A}XX^T\hat{A}/d_0$, then we have $\{H^{(1)}_{:,k}\}_{k\in[d_1]}$ are exactly i.i.d. from $N(\mathbf{0}_n, \Sigma^{(1)})$.

Induction step. Suppose that $\{H^{(l)}_{:,k}\}_{k\in[d_l]}$ converge to i.i.d. n-dimensional Gaussian random variables $N(0,\Sigma^{(l)})$ in distribution as $d_1,\dots,d_{l-1}$ sequentially go to infinity, we look at the $(l+1)$-th layer. Recall from the formation of the l-th layer in vanilla GCN, we have

$$\begin{aligned}H^{(l+1)} &= \hat{A}X^{(l)}W^{(l+1)} + \mathbf{1}_n\cdot b^{(l+1)},\\ X^{(l)} &= \sigma(H^{(l)}),\end{aligned}$$

for any $l\geq 1$. We vectorize the first equation and get

$$\begin{aligned}\mathrm{vec}(H^{(l+1)}) &= \mathrm{vec}(\hat{A}X^{(l)}W^{(l+1)}) + \mathrm{vec}(\mathbf{1}_n\cdot b^{(l+1)})\\ &= \sum_{k=1}^{d_l}\mathrm{vec}\left(\underbrace{[\hat{A}X^{(l)}_{:,k}]}_{n\times 1}\cdot\underbrace{W^{(l+1)}_{k,:}}_{1\times d_{l+1}}\right),\end{aligned} \tag{5}$$

because $b^{(l+1)}$ is initialized to be $\mathbf{0}_{d_{l+1}}$ under Assumption 3.1. Suppose that $\Sigma^{(l+1)} = \sigma_w^2\hat{A}G(\Sigma^{(l)})\hat{A}$, we only need to show that $\mathrm{vec}(H^{(l+1)})$ converges to a Gaussian random variable $N(\mathbf{0}_{nd_{l+1}}, I_{d_{l+1}}\otimes\Sigma^{(l+1)})$ in distribution as $d_1, d_2, \dots, d_{l-1}, d_l$ sequentially go to infinity.

For brevity, we define

$$\omega^{(l+1)}_{kk'} := \sqrt{d_l}\cdot W^{(l+1)}_{kk'},\quad \text{for all } k\in[d_l] \text{ and } k'\in[d_{l+1}],$$

and

$$Z_k^{(l+1)} := \operatorname{vec}\left([\hat{A}X_{:,k}^{(l)}] \cdot \omega_{k,:}^{(l+1)}\right), \quad \text{for all } k \in [d_l]. \tag{6}$$

Then we get that $\{\omega_{kk'}^{(l+1)}\}_{k\in[d_l],k'\in[d_{l+1}]}$ are i.i.d. from $N(0, \sigma_w^2)$ and

$$\text{RHS of (5)} = \frac{1}{\sqrt{d_l}} \sum_{k=1}^{d_l} Z_k^{(l+1)}. \tag{7}$$

By the induction hypothesis, as $d_1, d_2 \ldots, d_{l-1}$ sequentially go to infinity, $\{X_{:,k}^{(l)}\}_{k\in[d_l]} = \{\sigma(H_{:,k}^{(l)})\}_{k\in[d_l]}$ converge to i.i.d. n-dimensional random vectors in distribution. Because $X^{(l)}$ can be regarded as a function of $\{W^{(l')}\}_{l'=1}^{l}$ at initialization, we get that $X^{(l)}$ and $W^{(l+1)}$ are independent. Thus, as $d_1, d_2 \ldots, d_{l-1}$ sequentially go to infinity, $\{Z_k^{(l+1)}\}_{k\in[d_l]}$ converge to i.i.d. random vectors in distribution. Moreover, in this limiting case, by taking the expectation of (6), we have

$$\mathbb{E}[Z_1^{(l+1)}] = \operatorname{vec}\left(\left[\hat{A}\mathbb{E}[X_{:,k}^{(l)}]\right] \cdot \mathbb{E}[\omega_{k,:}^{(l+1)}]\right) = \operatorname{vec}\left(\mathbf{0}_{n\times 1} \cdot \mathbf{0}_{1\times d_{l+1}}\right) = \mathbf{0}_{nd_{l+1}}.$$

Calculating the covariance matrix of (6), we have

$$\begin{aligned}
\operatorname{Cov}[Z_1^{(l+1)}, Z_1^{(l+1)}] &= \mathbb{E}[Z_1^{(l+1)} \cdot Z_1^{(l+1)T}] \\
&= \mathbb{E}\left[\operatorname{vec}\left([\hat{A}X_{:,1}^{(l)}] \cdot \omega_{1,:}^{(l+1)}\right) \cdot \operatorname{vec}\left([\hat{A}X_{:,1}^{(l)}] \cdot \omega_{1,:}^{(l+1)}\right)^T\right] \\
&= \mathbb{E}\left[(\omega_{1,:}^{(l+1)T} \otimes \hat{A}X_{:,1}^{(l)}) \cdot (\omega_{1,:}^{(l+1)} \otimes X_{:,1}^{(l)T}\hat{A})\right] \\
&= \mathbb{E}\left[\omega_{1,:}^{(l+1)T}\omega_{1,:}^{(l+1)} \otimes \hat{A}X_{:,1}^{(l)}X_{:,1}^{(l)T}\hat{A}\right] \\
&= \mathbb{E}\left[\omega_{1,:}^{(l+1)T}\omega_{1,:}^{(l+1)}\right] \otimes \left\{\hat{A} \cdot \mathbb{E}\left[X_{:,1}^{(l)}X_{:,1}^{(l)T}\right] \cdot \hat{A}\right\} \\
&= \sigma_w^2 I_{d_{l+1}} \otimes \hat{A}G(\Sigma^{(l)})\hat{A} \\
&= I_{d_{l+1}} \otimes \sigma_w^2 \hat{A}G(\Sigma^{(l)})\hat{A} = I_{d_{l+1}} \otimes \Sigma^{(l+1)}.
\end{aligned}$$

Here $X_{:,1}^{(l)}$ actually stands for the limit of true $X_{:,1}^{(l)}$ as $d_1, \ldots, d_{l-1}$ sequentially go to infinity without bringing any confusion.

By multivariate central limit theorem, $\frac{1}{\sqrt{d_l}} \sum_{k=1}^{d_l} Z_k^{(l+1)}$ converges to a Gaussian random variable $N(\mathbf{0}_{nd_{l+1}}, I_{d_{l+1}} \otimes \Sigma^{(l+1)})$ in distribution as $d_l \to \infty$. Recalling (5) and (7), we conclude that $\operatorname{vec}(H^{(l+1)})$ converges to a Gaussian random variable $N(\mathbf{0}_{nd_{l+1}}, I_{d_{l+1}} \otimes \Sigma^{(l+1)})$ as $d_1, \ldots, d_l$ sequentially go to infinity.

Conclusion. By the principle of mathematical induction, we have proven this proposition. □

C.2. Some discussion w.r.t. G

We claim that the function G is well-defined in Proposition C.1 on the collection of positive semi-definite matrices

$$\mathcal{S} = \{\Sigma \in \mathbb{R}^{n\times n} : x^T\Sigma x \geq 0 \text{ for all } x \in \mathbb{R}^{n\times 1}\}. \tag{8}$$

Remark C.2. To show that $G(\Sigma) = \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\sigma(h)\sigma(h)^T]$ is well-defined at any $\Sigma \in \mathcal{S}$, we only need to show that such Σ is always a feasible covariance matrix of Gaussian distribution. For any $\Sigma \in \mathcal{S}$, there exists $P \in \mathbb{R}^{n\times n}$, such that $PP^T = \Sigma$. Let $\xi \sim N(\mathbf{0}_n, I_n)$ be an n-dimensional standard normal random variable, then the random variable $P\xi \sim N(\mathbf{0}_n, \Sigma)$. Thus, all positive semi-definite matrices are feasible covariance matrices for Gaussian distributions.

Definition C.3. Given any positive semi-definite matrix $\Sigma \in \mathcal{S}$, we define

$$G_1(\Sigma) := q(\Sigma)q(\Sigma)^T, \tag{9}$$

where $q(\Sigma) \in \mathbb{R}^{n\times 1}$ is defined by

$$q(\Sigma)_i := \sqrt{G(\Sigma)_{ii}}, \quad \text{for all } i \in [n]. \tag{10}$$

Lemma C.4. *Given any positive semi-definite matrix $\Sigma \in \mathcal{S}$, it holds that*

$$G_1(\Sigma)_{ij} \geq G(\Sigma)_{ij} \quad \textit{for any } i, j \in [n]. \tag{11}$$

Proof. Recalling the formation of function G in Proposition C.1 (NNGP correspondence for vanilla GCN), for any $i, j \in [n]$, we have

$$G(\Sigma)_{ij} = \mathbb{E}_{h \sim N(\mathbf{0}_n, \Sigma)}[\sigma(h_i) \cdot \sigma(h_j)].$$

Recalling (9) and (10) in Definition C.3, we get

$$\begin{aligned} G_1(\Sigma)_{ij} &:= q(\Sigma)_i \cdot q(\Sigma)_j = \sqrt{G(\Sigma)_{ii}} \cdot \sqrt{G(\Sigma)_{jj}} \\ &= \mathbb{E}_{h \sim N(\mathbf{0}_n, \Sigma)}[\sigma(h_i)^2]^{\frac{1}{2}} \cdot \mathbb{E}_{h \sim N(\mathbf{0}_n, \Sigma)}[\sigma(h_j)^2]^{\frac{1}{2}} \end{aligned} \tag{12}$$

From Hölder's inequality (Hardy et al., 1952), we get

$$\begin{aligned} \text{RHS of } (12) &\geq \mathbb{E}_{h \sim N(\mathbf{0}_n, \Sigma)} \left[|\sigma(h_i) \cdot \sigma(h_j)| \right] \\ &\geq \mathbb{E}_{h \sim N(\mathbf{0}_n, \Sigma)} \left[\sigma(h_i) \cdot \sigma(h_j) \right] = G(\Sigma)_{ij}. \end{aligned}$$

□

Lemma C.5. *Given the NNGP covariance matrices $\{\Sigma^{(l)}\}_{l=1}^{\infty}$ defined by (3), it holds that*

$$\operatorname{tr}(\Sigma^{(l+1)}) \leq \sigma_w^2 \operatorname{tr}(G(\Sigma^{(l)})).$$

Proof. Recalling the NNGP correspondence formula for vanilla GCN (3) in Proposition C.1, we have

$$\operatorname{tr}(\Sigma^{(l+1)}) = \operatorname{tr}(\sigma_w^2(\hat{A}G(\Sigma^{(l)})\hat{A})) = \sigma_w^2 \operatorname{tr}(\hat{A}G(\Sigma^{(l)})\hat{A})). \tag{13}$$

Since all entries of $\hat{A}$ are non-negative, by Lemma C.4, we have

$$(\hat{A}G(\Sigma^{(l)})\hat{A})_{ii} \leq (\hat{A}G_1(\Sigma^{(l)})\hat{A})_{ii}, \quad \text{for any } i \in [n].$$

Taking the summation of w.r.t $i \in [n]$, we get

$$\operatorname{tr}(\hat{A}G(\Sigma^{(l)})\hat{A}) \leq \operatorname{tr}(\hat{A}G_1(\Sigma^{(l)})\hat{A}). \tag{14}$$

Recalling the definition of function G_1 in (9), we get

$$\operatorname{tr}(\hat{A}G_1(\Sigma^{(l)})\hat{A}) = \operatorname{tr}(\hat{A}q(\Sigma^{(l)})q(\Sigma^{(l)})^T\hat{A}) = \|\hat{A}q(\Sigma^{(l)})\|^2. \tag{15}$$

By Proposition B.1, all the eigenvalues of $\hat{A}$ belong to $(-1, 1]$. Recalling the definition of function q in (10), we get

$$\|\hat{A}q(\Sigma^{(l)})\|^2 \leq \|q(\Sigma^{(l)})\|^2 = \sum_{i=1}^{n} q(\Sigma^{(l)})_i^2 = \operatorname{tr}(G(\Sigma^{(l)})). \tag{16}$$

Finally, combining (13), (14), (15), and (16), we complete the proof. □

C.3. Proof of Theorem 4.1 (signal propagation on ReLU-like-activated vanilla GCN)

We will give a more general signal propagation analysis on vanilla GCN with ReLU-like activation.

Definition C.6 (ReLU-like activation). An activation function $\sigma : \mathbb{R} \to \mathbb{R}$ is (α, β)-ReLU if it has the form

$$\sigma(x) = \begin{cases} \alpha x, & x \geq 0, \\ \beta x, & x < 0, \end{cases} \tag{17}$$

where $\alpha, \beta \in \mathbb{R}_+$ and not both of them are 0. We also call such σ a ReLU-like activation function.

Then we extend Theorem 4.1 from the special $(1, 0)$-ReLU-activated case to the general (α, β)-ReLU-activated case.

Theorem C.7. *Under Assumption 3.1 and the NNGP correspondence approximation, when the activation function σ is (α, β)-ReLU in Definition C.6, we have*

1. *The output diversity metric $\mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}[\mathrm{Dir}(H)/\|H\|_{\mathrm{F}}^2]$ is independent of the choice of σ_w^2.*
2. *When $\sigma_w^2 = 2/(\alpha^2 + \beta^2)$, either the output diversity metric*

$$\lim_{L\to\infty} \mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})} \left[\mathrm{Dir}(H)/\|H\|_{\mathrm{F}}^2\right] = 0,$$

or the forward propagation metric

$$\lim_{L\to\infty} \mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}[\|H\|_{\mathrm{F}}^2/\|X\|_{\mathrm{F}}^2] = 0.$$

3. *When $\sigma_w^2 < 2$, for any $L \geq 1$, the forward propagation metric satisfies*

$$\mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}[\|H\|_{\mathrm{F}}^2/\|X\|_{\mathrm{F}}^2] \leq \frac{2C}{(\alpha^2+\beta^2)d_0} \cdot \left(\frac{\sigma_w^2(\alpha^2+\beta^2)}{2}\right)^L.$$

We first prove part 1 of Theorem C.7 and leave the rest of proof in the end of this subsection.

Proof of part 1 in Theorem C.7 (the generalized version of Theorem 4.1). Under the Gaussian random initialization assumption 3.1 and the NNGP correspondence approximation, we only need to prove that

$$\frac{\Sigma^{(l)}(\sigma_w^2)}{\sigma_w^{2l}} = \frac{\Sigma^{(l)}(\tilde{\sigma}_w^2)}{\tilde{\sigma}_w^{2l}}, \quad \text{for any } l \geq 1 \text{ and } \sigma_w^2, \tilde{\sigma}_w^2 > 0. \tag{18}$$

If (18) holds, then $H \sim N(\mathbf{0}_n, \Sigma^{(L)}(\sigma_w^2))$ implies $\tilde{\sigma}_w^L H/\sigma_w^L \sim N(\mathbf{0}_n, \Sigma^{(L)}(\tilde{\sigma}_w^2))$. In this way, we have

$$\begin{aligned}\mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)}(\sigma_w^2))}\left[\frac{\mathrm{Dir}(H)}{\|H\|_F^2}\right] &= \mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)}(\sigma_w^2))}\left[\frac{\mathrm{Dir}(\tilde{\sigma}_w^L H/\sigma_w^L)}{\|\tilde{\sigma}_w^L H/\sigma_w^L\|_{\mathrm{F}}^2}\right] \\ &= \mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)}(\tilde{\sigma}_w^2))}\left[\frac{\mathrm{Dir}(H)}{\|H\|_F^2}\right].\end{aligned}$$

Now we prove (18) by mathematical induction. When $l = 1$, by Proposition C.1, we have

$$\frac{\Sigma^{(1)}(\sigma_w^2)}{\sigma_w^2} = \frac{1}{d_0}\hat{A}XX^T\hat{A} = \frac{\Sigma^{(1)}(\tilde{\sigma}_w^2)}{\tilde{\sigma}_w^2}, \quad \text{for any } \sigma_w^2, \tilde{\sigma}_w^2 > 0.$$

If (18) holds for L, we look at the case for $L + 1$. Since the activation σ is (α, β)-ReLU, for any $c \in \mathbb{R}_+$, we have $\sigma(cx) = c\sigma(x)$. Recalling the definition of G in Proposition C.1, for any positive semi-definite matrix $\Sigma \in \mathcal{S}$, we have

$$\begin{aligned}G(c^2\Sigma)_{ij} &= \mathbb{E}_{h\sim N(\mathbf{0}_n,c^2\Sigma)}[\sigma(h_i)\cdot\sigma(h_j)] = \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\sigma(ch_i)\cdot\sigma(ch_j)] \\ &= c^2\mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\sigma(h_i)\cdot\sigma(h_j)] = c^2G(\Sigma)_{ij},\end{aligned}$$

for any $i, j \in [n]$ and $c \in \mathbb{R}_+$. Thus, by Proposition C.1, we have

$$\begin{aligned}\left(\frac{\tilde{\sigma}_w^2}{\sigma_w^2}\right)^{L+1} \cdot \Sigma^{(L+1)}(\sigma_w^2) &\overset{(a)}{=} \left(\frac{\tilde{\sigma}_w^2}{\sigma_w^2}\right)^{L+1} \cdot \sigma_w^2\hat{A}G\left(\Sigma^{(L)}(\sigma_w^2)\right)\hat{A} \\ &= \tilde{\sigma}_w^2 \cdot \left(\frac{\tilde{\sigma}_w^2}{\sigma_w^2}\right)^{L} \cdot \hat{A}G\left(\Sigma^{(L)}(\sigma_w^2)\right)\hat{A} \\ &\overset{(b)}{=} \tilde{\sigma}_w^2 \cdot \hat{A}G\left(\Sigma^{(L)}(\tilde{\sigma}_w^2)\right) \\ &\overset{(c)}{=} \Sigma^{(L+1)}(\tilde{\sigma}_w^2),\end{aligned}$$

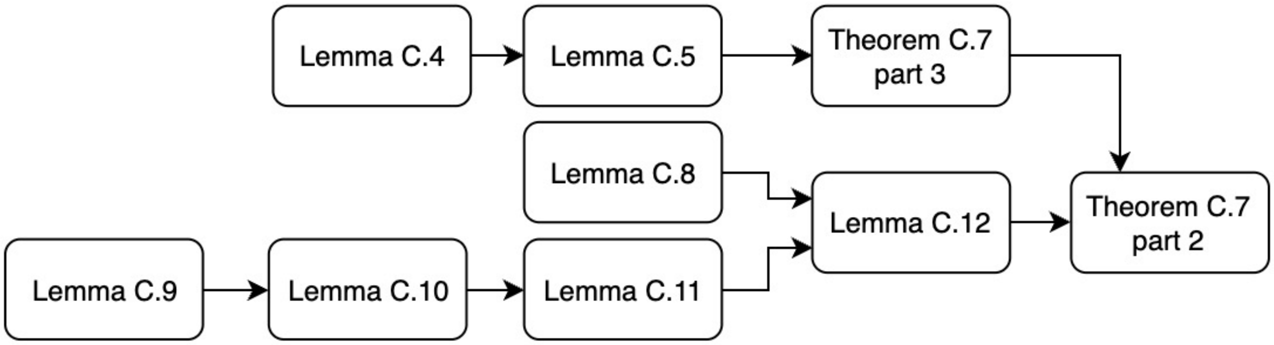


Figure 8. Roadmap for the proof of part 2 and 3 in Theorem C.7 (the generalized version of Theorem 4.1)

where (a) and (c) are due to Proposition C.1 and (b) are from the induction hypothesis.

Therefore, (18) holds for all $L \geq 1$ and we have completed the proof. □

For the convenience of following the proof of part 2 and 3, we provide a roadmap here.

Lemma C.8. *For any $x \in \mathbb{R}^n$, it holds that*

$$\mathrm{Dir}(\hat{A}x) \leq \lambda^2 \mathrm{Dir}(x), \tag{19}$$

where λ is the second largest absolute eigenvalue of $\hat{A}$, i.e.,

$$\lambda = \max_{i \in [n], \lambda_i \neq 1} |\lambda_i|.$$

Proof. Since $\hat{A}$ is a symmetric real matrix, by Proposition B.1, it can be decomposed as $\hat{A} = U\Lambda U^T$, where $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ and $U \in \mathbb{R}^{n\times n}$ is an orthogonal matrix. The i-th column u_i of U is the eigenvector corresponding to λ_i.

By Proposition B.1, we have $\lambda_i \in (-1, 1]$ for all $i \in [n]$. Since $\hat{L} = I - \hat{A}$, we conclude that

$$\begin{aligned}\mathrm{Dir}(\hat{A}x) = (\hat{A}x)^T\hat{L}\hat{A}x = x^T\hat{A}\hat{L}\hat{A}x &= z^TU^T(U\Lambda U^{-1})(U(I-\Lambda)U^{-1})(U\Lambda U^{-1})z \\ &= z^T\Lambda(I-\Lambda)\Lambda z = \sum_{i=1}^n (1-\lambda_i)\lambda_i^2 z_i^2 \leq \lambda^2 \sum_{i=1}^n (1-\lambda_i) z_i^2 \\ &= \lambda^2 z^T(I-\Lambda)z = \lambda^2 \mathrm{Dir}(x).\end{aligned}$$

□

Lemma C.9. *When the activation function σ is (α, β)-ReLU, it holds that*

$$(\sigma(x) - \sigma(y))^2 + (\sigma(-x) - \sigma(-y))^2 \leq (\alpha^2 + \beta^2)(x-y)^2, \tag{20}$$

for any $x, y \in \mathbb{R}$. Moreover, the inequality becomes an equality if and only if $xy \geq 0$.

Proof. When $x, y \geq 0$, it holds that

$$\text{LHS of (20)} = (\alpha x - \alpha y)^2 + (-\beta x + \beta y)^2 = \text{RHS of (20)}.$$

Similarly, the equality holds when $x, y \leq 0$. When $xy < 0$,

$$\begin{aligned}\text{LHS of (20)} &= (\alpha x - \beta y)^2 + (-\beta x + \alpha y)^2 \\ &= (\alpha^2 + \beta^2)(x^2 + y^2) - 4\alpha\beta xy \\ &= (\alpha^2 + \beta^2)(x - y)^2 + 2(\alpha - \beta)^2 xy \\ &< \text{RHS of (20)}.\end{aligned}$$

□

Lemma C.10. *When the activation function σ is (α, β)-ReLU, it holds that*

$$\mathrm{Dir}(\sigma(h)) + \mathrm{Dir}(\sigma(-h)) \leq (\alpha^2 + \beta^2)\mathrm{Dir}(h). \tag{21}$$

Proof. Since the activation function σ is (α, β)-ReLU, we have

$$\sigma(cx) = c\sigma(x), \quad \text{for any } c \in \mathbb{R}_+, x \in \mathbb{R}. \tag{22}$$

Then we get

$$\begin{aligned} \text{LHS of (21)} &= \sum_{(i,j)\in\mathcal{E}} \left[\frac{\sigma(h_i)}{\sqrt{1+d_i}} - \frac{\sigma(h_j)}{\sqrt{1+d_j}}\right]^2 + \left[\frac{\sigma(-h_i)}{\sqrt{1+d_i}} - \frac{\sigma(-h_j)}{\sqrt{1+d_j}}\right]^2 \\ &= \sum_{(i,j)\in\mathcal{E}} \left[\sigma\left(\frac{h_i}{\sqrt{1+d_i}}\right) - \sigma\left(\frac{h_j}{\sqrt{1+d_j}}\right)\right]^2 + \left[\sigma\left(\frac{-h_i}{\sqrt{1+d_i}}\right) - \sigma\left(\frac{-h_j}{\sqrt{1+d_j}}\right)\right]^2. \end{aligned} \tag{23}$$

By Lemma C.9, we have

$$\text{LHS of (21)} \leq (\alpha^2 + \beta^2) \sum_{(i,j)\in\mathcal{E}} \left[\frac{h_i}{\sqrt{1+d_i}} - \frac{h_j}{\sqrt{1+d_j}}\right]^2 = \text{RHS of (21)}. \tag{24}$$

□

Lemma C.11. *When the activation function σ is (α, β)-ReLU, for any feasible covariance matrix $\Sigma \in \mathbb{R}^{n\times n}$, it holds that*

$$\mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\mathrm{Dir}(\sigma(h))] \leq \frac{\alpha^2+\beta^2}{2} \cdot \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\mathrm{Dir}(h)].$$

Proof. By symmetry, for any n-dimensional random variable $h \sim N(\mathbf{0}_n, \Sigma)$, it holds that $-h \sim N(\mathbf{0}_n, \Sigma)$. By Lemma C.10, we have

$$\begin{aligned} 2\mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\mathrm{Dir}(\sigma(h))] &= \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\mathrm{Dir}(\sigma(h)) + \mathrm{Dir}(\sigma(-h))] \\ &\leq (\alpha^2+\beta^2)\mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\mathrm{Dir}(h)]. \end{aligned} \tag{25}$$

□

Lemma C.12. *Under Assumption 3.1 and the NNGP correspondence approximation, suppose that the activation function σ is (α, β)-ReLU in Definition C.6. If*

$$\sigma_w^2 < \frac{2}{\lambda^2(\alpha^2+\beta^2)},$$

then we have

$$\mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma^{(l)})}[\mathrm{Dir}(h)] = O\left(\left(\frac{\lambda^2\sigma_w^2(\alpha^2+\beta^2)}{2}\right)^l\right), \quad \textit{as } l \to \infty, \tag{26}$$

where λ is the second largest non-one absolute eigenvalue of $\hat{A}$, i.e.,

$$\lambda = \max_{i\in[n],\lambda_i\neq 1} |\lambda_i|.$$

Proof. For any positive semi-definite matrix $\Sigma \in \mathcal{S}$ and any n-dimensional Gaussian random variable $h \sim N(\mathbf{0}_n, \Sigma)$, we have

$$\mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\mathrm{Dir}(h)] = \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\mathrm{tr}(h^T\hat{L}h)] = \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\mathrm{tr}(\hat{L}hh^T)] = \mathrm{tr}(\hat{L}\Sigma).$$

Then according the NNGP correspondence formula (3) in Proposition C.1, for any $l \in \mathbb{N}$, we have

$$\begin{aligned}
&\mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma^{(l+1)})}[\mathrm{Dir}(h)] = \mathrm{tr}(\hat{L}\Sigma^{(l+1)}) \\
&= \sigma_w^2 \,\mathrm{tr}(\hat{L}\hat{A}G(\Sigma^{(l)})\hat{A}) = \sigma_w^2 \,\mathrm{tr}\left(\hat{L}\hat{A}\cdot \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma^{(l)})}[\sigma(h)\sigma(h)^T]\cdot \hat{A}\right) \\
&= \sigma_w^2 \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma^{(l)})}\left[\mathrm{tr}\left(\hat{L}\hat{A}\sigma(h)\sigma(h)^T\hat{A}\right)\right] = \sigma_w^2 \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma^{(l)})}\left[\mathrm{tr}\left(\sigma(h)^T\hat{A}\hat{L}\hat{A}\sigma(h)\right)\right] \\
&= \sigma_w^2 \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma^{(l)})}\left[\mathrm{Dir}\left(\hat{A}\sigma(h)\right)\right].
\end{aligned} \tag{27}$$

By Lemma C.8 and Lemma C.11, we get

$$\text{RHS of (27)} \le \lambda^2\sigma_w^2 \cdot \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma^{(l)})}[\mathrm{Dir}(\sigma(h))] \le \frac{\lambda^2\sigma_w^2(\alpha^2+\beta^2)}{2}\cdot \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma^{(l)})}[\mathrm{Dir}(h)]. \tag{28}$$

Thus, combining (27) and (28), by induction, we have

$$\mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma^{(l)})}[\mathrm{Dir}(h)] = O\left(\left(\frac{\lambda^2\sigma_w^2(\alpha^2+\beta^2)}{2}\right)^l\right), \quad \text{as } l\to\infty.$$

□

Proof of part 2 *and* 3 *in Theorem C.7 (the generalized version of Theorem 4.1).* First of all, we will prove **part 3** of this theorem. For any positive semi-definite matrix $\Sigma \in \mathcal{S}$, we have

$$\mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\|h\|^2] = \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\mathrm{tr}(h^Th)] = \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\mathrm{tr}(hh^T)] = \mathrm{tr}(\Sigma).$$

For this reason, we only need to focus on $\{\mathrm{tr}(\Sigma^{(l)})\}_{l=1}^{\infty}$ in the following proof.

We will show that $\{\mathrm{tr}(\Sigma^{(l)})\}_{l=1}^{\infty}$ is a decreasing sequence if $\sigma_w \le 2/(\alpha^2+\beta^2)$. By Lemma C.5, we have

$$\mathrm{tr}(\Sigma^{(l+1)}) \le \sigma_w^2 \,\mathrm{tr}(G(\Sigma^{(l)})). \tag{29}$$

When the activation function σ is (α,β)-ReLU, for any $c\in\mathbb{R}_+$, it holds that

$$\begin{aligned}
\mathbb{E}_{Z\sim N(0,1)}[\sigma(cZ)^2] &= \mathbb{E}_{Z\sim N(0,1)}[\alpha^2c^2Z^2\mathbb{1}_{\{Z>0\}}] + \mathbb{E}_{Z\sim N(0,1)}[\beta^2c^2Z^2\mathbb{1}_{\{Z\le 0\}}] \\
&= \frac{\alpha^2+\beta^2}{2}\cdot \mathbb{E}_{Z\sim N(0,1)}[c^2Z^2].
\end{aligned}$$

Accordingly, for any positive semi-definite matrix $\Sigma\in\mathcal{S}$ and $i\in[n]$, we have

$$\begin{aligned}
G(\Sigma)_{ii} &= \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\sigma(h_i)^2] = \mathbb{E}_{Z\sim N(0,1)}\left[\sigma(\sqrt{\Sigma_{ii}}Z)^2\right] \\
&= \frac{\alpha^2+\beta^2}{2}\cdot \mathbb{E}_{Z\sim N(0,1)}\left[\Sigma_{ii}Z^2\right] = \frac{\alpha^2+\beta^2}{2}\cdot\Sigma_{ii}.
\end{aligned} \tag{30}$$

Combining (29) and (30), we get

$$\mathrm{tr}(\Sigma^{(l+1)}) \le \frac{\sigma_w^2(\alpha^2+\beta^2)}{2}\,\mathrm{tr}(\Sigma^{(l)}).$$

Thus, we have shown that $\{\mathrm{tr}(\Sigma^{(l)})\}_{l=1}^{\infty}$ is a decreasing sequence if $\sigma_w \le 2/(\alpha^2+\beta^2)$. In addition, if $\sigma_w < 2/(\alpha^2+\beta^2)$, we get

$$\mathrm{tr}(\Sigma^{(L)}) \le \left(\frac{\sigma_w^2(\alpha^2+\beta^2)}{2}\right)^{L-1}\mathrm{tr}(\Sigma^{(1)}). \tag{31}$$

By Proposition C.1, we have

$$\mathrm{tr}(\Sigma^{(1)}) = \frac{\sigma_w^2}{d_0}\,\mathrm{tr}(\hat{A}XX^T\hat{A}) = \frac{\sigma_w^2}{d_0}\sum_{k=1}^{d_0}\mathrm{tr}(\hat{A}X_{:,k}X_{:,k}^T\hat{A}) = \frac{\sigma_w^2}{d_0}\sum_{k=1}^{d_0}\|\hat{A}X_{:,k}\|^2 \tag{32}$$

Since all the eigenvalues of $\hat{A}$ belong to $(-1,1]$ by Propositon B.1, we get

$$\text{RHS of (32) } \leq \frac{\sigma_w^2}{d_0}\sum_{k=1}^{d_0}\|X_{:,k}\|^2 = \frac{\sigma_w^2}{d_0}\operatorname{tr}(XX^T). \tag{33}$$

Combining (31), (32), and (33), we have

$$\operatorname{tr}(\Sigma^{(L)}) \leq \frac{\sigma_w^2}{d_0}\cdot\left(\frac{\sigma_w^2(\alpha^2+\beta^2)}{2}\right)^{L-1}\operatorname{tr}(XX^T).$$

Thus, the forward propagation metric at the L-th layer satisfies

$$\begin{aligned}\mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}\left[\frac{\|H\|_{\mathrm{F}}^2}{\|X\|_{\mathrm{F}}^2}\right] &= \frac{C}{\operatorname{tr}(XX^T)}\cdot\mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma^{(L)})}[\|h\|^2] = \frac{C}{\operatorname{tr}(XX^T)}\operatorname{tr}(\Sigma^{(L)})\\ &\leq \frac{C\sigma_w^2}{d_0}\cdot\left(\frac{\sigma_w^2(\alpha^2+\beta^2)}{2}\right)^{L-1}\\ &= \frac{2C}{(\alpha^2+\beta^2)d_0}\cdot\left(\frac{\sigma_w^2(\alpha^2+\beta^2)}{2}\right)^{L}.\end{aligned}$$

Then we have completed part 3 of this theorem. If σ is ReLU activation function, i.e., $(1,0)$-ReLU. If $\sigma < 2 = \frac{2}{1^2+0^2}$, we have

$$\mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}\left[\frac{\|H\|_{\mathrm{F}}^2}{\|X\|_{\mathrm{F}}^2}\right] = \frac{2C}{(1^2+0^2)d_0}\cdot\left(\frac{\sigma_w^2(1^2+0^2)}{2}\right)^{L} = \frac{2C}{d_0}\cdot\left(\frac{\sigma_w^2}{2}\right)^{L},$$

which coincides with part 3 in Theorem 4.1.

Next, we will prove **part 2** of this theorem. Let's study the case when $\sigma_w = 2/(\alpha^2+\beta^2)$. Suppose that

$$\lim_{l\to\infty}\operatorname{tr}(\Sigma^{(l)}) = \delta_0.$$

If $\delta_0 = 0$, then we have completed the first part of this theorem by getting

$$\begin{aligned}\lim_{L\to\infty}\mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}[\|H\|_{\mathrm{F}}^2/\|X\|_{\mathrm{F}}^2] &= \lim_{L\to\infty}\frac{C}{\|X\|_{\mathrm{F}}^2}\cdot\mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma^{(L)})}[\|h\|^2]\\ &= \frac{C}{\|X\|_{\mathrm{F}}^2}\cdot\lim_{L\to\infty}\operatorname{tr}(\Sigma^{(L)}) = 0.\end{aligned}$$

Now we study the case when $\delta_0 > 0$. In order to show part 2 of the theorem, we only need to demonstrate that

$$\lim_{L\to\infty}\mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}\left[\frac{\operatorname{Dir}(H)}{\|H\|_{\mathrm{F}}^2}\right] = 0.$$

Given any fixed $\epsilon > 0$, we have

$$\begin{aligned}&\mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}\left[\frac{\operatorname{Dir}(H)}{\|H\|_{\mathrm{F}}^2}\right]\\ &= \mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}\left[\frac{\operatorname{Dir}(H)}{\|H\|_{\mathrm{F}}^2}\mathbb{1}_{\{\|H\|_{\mathrm{F}}\geq\epsilon\}}\right] + \mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}\left[\frac{\operatorname{Dir}(H)}{\|H\|_{\mathrm{F}}^2}\mathbb{1}_{\{\|H\|_{\mathrm{F}}\leq\epsilon\}}\right].\end{aligned} \tag{34}$$

From Lemma B.2, it holds that $\operatorname{Dir}(H)/\|H\|_{\mathrm{F}}^2 \leq 2$, so we get

$$\begin{aligned}\text{RHS of (34) } &\leq \frac{1}{\epsilon^2}\cdot\mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}\left[\operatorname{Dir}(H)\mathbb{1}_{\{\|H\|_{\mathrm{F}}\geq\epsilon\}}\right] + 2\cdot\mathbb{P}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}\left[\|H\|_{\mathrm{F}}\leq\epsilon\right]\\ &\leq \frac{1}{\epsilon^2}\cdot\mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}\left[\operatorname{Dir}(H)\right] + 2\cdot\mathbb{P}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}\left[\|H\|_{\mathrm{F}}\leq\epsilon\right].\end{aligned} \tag{35}$$

For any $L \geq 1$, there exists $i \in [n]$, such that $\Sigma_{ii}^{(L)} \geq \operatorname{tr}(\Sigma^{(L)})/n$. Then for any $n \times C$ random matrix $H \sim N(\mathbf{0}_n, \Sigma^{(L)})$, we have $H_{i,1} \sim N(0, \Sigma_{ii}^{(L)})$. For this reason, we have

$$\begin{aligned}
\mathbb{P}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}\left[\|H\|_{\mathrm{F}} \leq \epsilon\right] \leq \mathbb{P}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}\left[|H_{i,1}| \leq \epsilon\right] &= \mathbb{P}_{Z\sim N(0,1)}\left[|Z| \leq \frac{\epsilon}{\sqrt{\Sigma_{ii}}}\right] \\
&\leq \mathbb{P}_{Z\sim N(0,1)}\left[|Z| \leq \epsilon \cdot \sqrt{\frac{n}{\operatorname{tr}(\Sigma^{(L)})}}\right] \\
&= 2\Phi\left(\epsilon \cdot \sqrt{\frac{n}{\operatorname{tr}(\Sigma^{(L)})}}\right) - 1,
\end{aligned} \tag{36}$$

where $\Phi(x) = \mathbb{P}_{Z\sim N(0,1)}[Z \leq x]$ denotes the cumulative distribution function of the standard normal distribution $N(0,1)$.

Combining (34), (35), and (36), we get

$$\mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}\left[\frac{\operatorname{Dir}(H)}{\|H\|_{\mathrm{F}}^2}\right] \leq \frac{1}{\epsilon^2} \cdot \mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}\left[\operatorname{Dir}(H)\right] + 4\Phi\left(\epsilon \cdot \sqrt{\frac{n}{\operatorname{tr}(\Sigma^{(L)})}}\right) - 2,$$

for any $L \geq 1$.

Since

$$\sigma_w^2 = \frac{2}{\alpha^2 + \beta^2} < \frac{2}{\lambda^2(\alpha^2 + \beta^2)},$$

by Lemma C.12, we have

$$\lim_{L\to\infty} \mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}[\operatorname{Dir}(H)] = 0.$$

We let $L \to \infty$ in (34) and get

$$\begin{aligned}
&\limsup_{L\to\infty} \mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}\left[\frac{\operatorname{Dir}(H)}{\|H\|_{\mathrm{F}}^2}\right] \\
\leq\ & \frac{1}{\epsilon^2} \cdot \limsup_{L\to\infty} \mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}\left[\operatorname{Dir}(H)\right] + 4 \cdot \limsup_{L\to\infty} \Phi\left(\epsilon \cdot \sqrt{\frac{n}{\operatorname{tr}(\Sigma^{(L)})}}\right) - 2 \\
=\ & \frac{1}{\epsilon^2} \cdot 0 + 4\Phi\left(\epsilon \cdot \sqrt{\frac{n}{\delta_0}}\right) - 2 = 4\Phi\left(\epsilon \cdot \sqrt{\frac{n}{\delta_0}}\right) - 2.
\end{aligned} \tag{37}$$

Notice that the left hand side of (37) is independent of the choice of ϵ. Since Φ is a continuous map, we let $\epsilon \to 0^+$ and get

$$\limsup_{L\to\infty} \mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}\left[\frac{\operatorname{Dir}(H)}{\|H\|_{\mathrm{F}}^2}\right] \leq 4\Phi(0) - 2 = 0.$$

Therefore, we have

$$\lim_{L\to\infty} \mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}\left[\frac{\operatorname{Dir}(H)}{\|H\|_{\mathrm{F}}^2}\right] = 0.$$

□

C.4. Proof of Theorem 4.2 (signal propagation on tanh-activated vanilla GCN)

Lemma C.13. *The collection of positive semi-definite matrices $\mathcal{S}$ defined by (8) is a closed subset of $\mathbb{R}^{n\times n}$.*

Proof. We only need to show that given any convergent sequence $\{Q^{(k)}\}_{k=1}^{\infty} \subset \mathcal{S}$, its limit also belongs to $\mathcal{S}$. Suppose that

$$\lim_{k\to\infty} Q^{(k)} = Q^*.$$

Since all $\{Q^{(k)}\}_{k=1}^{\infty}$ are positive semi-definite matrices, so given any $x \in \mathbb{R}^{n\times 1}$, we have

$$x^T Q^{(k)} x \geq 0, \quad \text{for all } k \in \mathbb{N}.$$

Then we get

$$x^T Q^* x = \lim_{k\to\infty} x^T Q^{(k)} x \geq 0.$$

Thus, Q^* also belongs to $\mathcal{S}$. □

Lemma C.14. *When the activation function σ is tanh, i.e., $\sigma(x) = (e^x - e^{-x})/(e^x + e^{-x})$, then we have $|\sigma(x)| \leq |x|$ for any $x \in \mathbb{R}$. Moreover, the equality holds if and only if $x = 0$.*

Proof. It is easy to verify that $\sigma(0) = 0$. Given any $x \geq 0$, we have

$$\sigma(-x) = \frac{e^{-x} - e^x}{e^{-x} + e^x} = -\frac{e^x - e^{-x}}{e^{-x} + e^x} = -\sigma(x).$$

For this reason, we only need to prove that $|\sigma(x)| < |x|$ for any $x > 0$. In the following part, we will show that $0 < \sigma(x) < x$ when $x > 0$.

We define $f(x) := \sigma(x) - x$ for any $x \geq 0$. Let's consider the derivative of f:

$$\begin{aligned} f'(x) &= \frac{d}{dx}\left(\frac{e^x - e^{-x}}{e^x + e^{-x}} - x\right) \\ &= \frac{1}{(e^x + e^{-x})^2}\left[(e^x + e^{-x}) \cdot \frac{d}{dx}(e^x - e^{-x}) - (e^x - e^{-x}) \cdot \frac{d}{dx}(e^x + e^{-x})\right] - 1 \\ &= \frac{(e^x + e^{-x})^2 - (e^x - e^{-x})^2}{(e^x + e^{-x})^2} - 1 \\ &= \frac{-(e^x - e^{-x})^2}{(e^x + e^{-x})^2}. \end{aligned}$$

Then if $x > 0$, we have $f'(x) < 0$; if $x = 0$, we have $f'(x) = 0$. Thus, $f(x) = \sigma(x) - x$ is a strictly decreasing function in $[0, +\infty)$. Since $f(0) = \sigma(0) - 0 = 0$, we have

$$f(x) = \sigma(x) - x < 0, \quad \text{for any } x > 0.$$

Since $0 < e^x - e^{-x} < e^x + e^{-x}$ for any $x > 0$, it holds that

$$\sigma(x) = (e^x - e^{-x})/(e^x + e^{-x}) > 0, \quad \text{for any } x > 0.$$

Therefore, we get that $0 < \sigma(x) < x$ for any $x > 0$ and have completed the proof of this lemma. □

Now it is time for Theorem 4.2.

Proof of Theorem 4.2. First of all, we will prove **part 2** of this theorem. For any positive semi-definite matrix $\Sigma \in \mathcal{S}$, we have

$$\mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\|h\|^2] = \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\operatorname{tr}(h^T h)] = \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\operatorname{tr}(hh^T)] = \operatorname{tr}(\Sigma).$$

For this reason, we only need to focus on $\{\operatorname{tr}(\Sigma^{(l)})\}_{l=1}^{\infty}$ in the following proof.

We will show that $\{\operatorname{tr}(\Sigma^{(l)})\}_{l=1}^{\infty}$ is a decreasing sequence if $\sigma_w \leq 1$. By Lemma C.5, we have

$$\operatorname{tr}(\Sigma^{(l+1)}) \leq \sigma_w^2 \operatorname{tr}(G(\Sigma^{(l)})). \tag{38}$$

By Lemma C.14, we have $|\sigma(x)| \leq |x|$ for any $x \in \mathbb{R}$. Moreover, the equality holds if and only if $x = 0$. For this reason, given any positive semi-definite matrix $\Sigma \in \mathcal{S}$, we have

$$\begin{aligned} \operatorname{tr}(G(\Sigma)) &= \sum_{i=1}^n \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\sigma(h_i)^2] = \sum_{i=1}^n \mathbb{E}_{Z\sim N(0,1)}\left[\sigma(\sqrt{\Sigma_{ii}}Z)^2\right] \\ &\leq \sum_{i=1}^n \mathbb{E}_{Z\sim N(0,1)}\left[(\sqrt{\Sigma_{ii}}Z)^2\right] = \sum_{i=1}^n \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[h_i^2] = \operatorname{tr}(\Sigma), \end{aligned} \tag{39}$$

and the inequality becomes an equality if and only if $\sqrt{\Sigma_{ii}}Z = 0$ holds $\mathbb{P}$-a.s. for all $i \in [n]$. Since $Z \sim N(0,1)$ follows a standard normal distribution, it is equivalent to $\Sigma_{ii} = 0$ for all $i \in [n]$, i.e., $\operatorname{tr}(\Sigma) = 0$.

Combining (38) and (39), we get

$$\operatorname{tr}(\Sigma^{(l+1)}) \le \sigma_w^2 \operatorname{tr}(\Sigma^{(l)}).$$

Thus, we have shown that $\{\operatorname{tr}(\Sigma^{(l)})\}_{l=1}^{\infty}$ is a decreasing sequence if $\sigma_w \le 1$. In addition, if $\sigma_w < 1$, we get

$$\operatorname{tr}(\Sigma^{(L)}) \le \sigma_w^{2(L-1)} \operatorname{tr}(\Sigma^{(1)}). \tag{40}$$

Analogous to the proof of part 3 in Theorem C.7 for ReLU-activated model, by Proposition C.1 and Proposition B.1, we have

$$\begin{aligned} \operatorname{tr}(\Sigma^{(1)}) = \frac{\sigma_w^2}{d_0} \operatorname{tr}(\hat{A}XX^T\hat{A}) &= \frac{\sigma_w^2}{d_0} \sum_{k=1}^{d_0} \operatorname{tr}(\hat{A}X_{:,k}X_{:,k}^T\hat{A}) \\ &= \frac{\sigma_w^2}{d_0} \sum_{k=1}^{d_0} \|\hat{A}X_{:,k}\|^2 \le \frac{\sigma_w^2}{d_0} \sum_{k=1}^{d_0} \|X_{:,k}\|^2 = \frac{\sigma_w^2}{d_0} \|X\|_{\mathrm{F}}^2. \end{aligned} \tag{41}$$

Combining (40) and (41), we have

$$\operatorname{tr}(\Sigma^{(1)}) \le \frac{\sigma_w^{2L}}{d_0} \|X\|_{\mathrm{F}}^2.$$

Then we have completed part 2 of the theorem by getting

$$\begin{aligned} \mathbb{E}_{H \sim N(\mathbf{0}_n, \Sigma^{(L)})}\left[\frac{\|H\|_{\mathrm{F}}^2}{\|X\|_{\mathrm{F}}^2}\right] &= \frac{C}{\|X\|_{\mathrm{F}}^2} \mathbb{E}_{h \sim N(\mathbf{0}_n, \Sigma^{(L)})}[\|h\|^2] = \frac{C}{\|X\|_{\mathrm{F}}^2} \operatorname{tr}(\Sigma^{(L)}) \\ &\le \frac{C}{\|X\|_{\mathrm{F}}^2} \cdot \frac{\sigma_w^{2L}}{d_0} \cdot \|X\|_{\mathrm{F}}^2 \le \frac{C}{d_0} \cdot \sigma_w^{2L}. \end{aligned}$$

Next, we will prove **part 1** of this theorem. Let's study the case when $\sigma_w = 1$.

Since $\Sigma^{(l)}$ is a positive semi-definite matrix for any $l \in \mathbb{N}$, we have

$$|\Sigma_{ij}^{(l)}|^2 \le \Sigma_{ii}^{(l)}\Sigma_{jj}^{(l)} \le \operatorname{tr}(\Sigma^{(l)})^2 \le \operatorname{tr}(\Sigma^{(1)})^2, \quad \text{for all } i,j \in [n].$$

Taking the summation of both sides w.r.t. i and j, we get

$$\|\Sigma^{(l)}\|_F^2 = \sum_{i,j=1}^{n} |\Sigma_{ij}^{(l)}|^2 \le n^2 \operatorname{tr}(\Sigma^{(1)})^2 < \infty.$$

Thus, the matrix sequence $\{\Sigma^{(l)}\}_{l=1}^{\infty}$ lies in

$$\mathcal{S}' = \mathcal{S} \cap \{\Sigma \in \mathbb{R}^{n\times n} : \|\Sigma\|_F \le n \operatorname{tr}(\Sigma^{(1)})\}.$$

By Lemma C.13, $\mathcal{S}'$ is a bounded and closed subset, i.e., a compact subset, of $\mathbb{R}^{n\times n}$. By the Bolzano–Weierstrass theorem, there exists a subsequence $\{\Sigma^{(l_k)}\}_{k=1}^{\infty}$ of $\{\Sigma^{(l)}\}_{l=1}^{\infty}$ and $\Sigma^* \in \mathcal{S}'$ such that

$$\lim_{k\to\infty} \Sigma^{(l_k)} = \Sigma^*.$$

Recalling (38) and that $\{\operatorname{tr}(\Sigma^{(l)})\}_{l=1}^{\infty}$ is a decreasing sequence, we have

$$\operatorname{tr}(\Sigma^{(l_{k+1})}) \le \operatorname{tr}(\Sigma^{(l_k+1)}) \le \operatorname{tr}(G(\Sigma^{(l_k)})).$$

Since G is a continuous function, we let $k \to \infty$ and get

$$\operatorname{tr}(\Sigma^*) = \lim_{k\to\infty} \operatorname{tr}(\Sigma^{(l_{k+1})}) \le \lim_{k\to\infty} \operatorname{tr}(G(\Sigma^{(l_k)})) = \operatorname{tr}(G(\Sigma^*)).$$

According to (39), we have

$$\operatorname{tr}(G(\Sigma^*)) = \operatorname{tr}(\Sigma^*).$$

This implies $\operatorname{tr}(\Sigma^*) = 0$ by (39).

Then, since $\{\operatorname{tr}(\Sigma^{(l)})\}_{l=1}^{\infty}$ is a decreasing sequence, we have

$$\lim_{l\to\infty} \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma^{(l)})}[\|h\|^2] = \lim_{l\to\infty} \operatorname{tr}(\Sigma^{(l)}) = \lim_{k\to\infty} \operatorname{tr}(\Sigma^{(l_k)}) = \operatorname{tr}(\Sigma^*) = 0.$$

Consequently, we have

$$\lim_{L\to\infty} \mathbb{E}_{H\sim N(\mathbf{0}_n,\Sigma^{(L)})}\left[\frac{\|H\|_{\mathrm{F}}^2}{\|X\|_{\mathrm{F}}^2}\right] = \frac{C}{\|X\|_{\mathrm{F}}^2} \lim_{L\to\infty} \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma^{(L)})}[\|h\|^2] = 0.$$

□

D. Signal propagation theory for linear ResGCN

D.1. NNGP correspondence for linear ResGCN

Proposition D.1 (NNGP correspondence for linear ResGCN)**.** *Under the initialization in Assumption 3.1, as the width of the hidden layers $d \to \infty$, the l-th layer's post-activation embedding channels $\{X^{(l)}_{:,k}\}_{k\in[d]}$ converge to i.i.d. Gaussian random variables $N(\mathbf{0}_n, \tilde{\Sigma}^{(l)})$ in distribution. The covariance matrices are*

$$\begin{aligned}\tilde{\Sigma}^{(0)} &= \frac{\sigma_w^2}{d_0} X X^T, \\ \tilde{\Sigma}^{(l+1)} &= \sigma_w^2 \alpha^2 \hat{A} \tilde{\Sigma}^{(l)} \hat{A} + \beta^2 \tilde{\Sigma}^{(l)}.\end{aligned} \tag{42}$$

Moreover, for an L-layer linear ResGCN, all the channels of $H^{(\mathrm{out},L)}$ converge to i.i.d. Gaussian random variables $N(\mathbf{0}_n, \tilde{\Sigma}^{(\mathrm{out},L)})$ in distribution, where

$$\tilde{\Sigma}^{(\mathrm{out},L)} = \sigma_w^2 \tilde{\Sigma}^{(L)}. \tag{43}$$

Proof of Proposition D.1. In linear ResGCN defined in Section 3.2, the post-activation embeddings $\{X^{(l)}\}_{l=1}^L$ satisfy

$$\begin{aligned}X^{(0)} &= XW^{(0)} + \mathbf{1}_n \cdot b^{(0)}, \\ X^{(l)} &= \alpha\sigma(H^{(l)}) + \beta X^{(l-1)} = \alpha(\hat{A} X^{(l-1)} W^{(l)} + \mathbf{1}_n \cdot b^{(l)}) + \beta X^{(l-1)}.\end{aligned} \tag{44}$$

Similar to the proof of Proposition C.1, We will prove the **first part** of Proposition D.1 by mathematical induction.

Base case. Under the initialization in Assumption 3.1, when $l = 0$, the k-th channel of $X^{(0)}$ is

$$X^{(0)}_{:,k} = XW^{(0)}_{:,k} + \mathbf{1}_n \cdot b^{(0)}_k = XW^{(0)}_{:,k}. \tag{45}$$

According to Assumption 3.1, the weights $\{W^{(0)}_{:,k}\}_{k\in[d]}$ are i.i.d. Gaussian distributed, so $\{X^{(0)}_{:,k}\}_{k\in[d]}$ are also i.i.d. Gaussian distributed. Taking the expectation of (45), we get

$$\mathbb{E}[X^{(0)}_{:,k}] = X \cdot \mathbb{E}[W^{(0)}_{:,k}] = 0.$$

Calculating the covariance matrix of (45), we have

$$\begin{aligned}\mathrm{Cov}[X^{(0)}_{:,k}, X^{(0)}_{:,k}] &= \mathbb{E}[X^{(0)}_{:,k} \cdot X^{(0)T}_{:,k}] = \mathbb{E}[XW^{(0)}_{:,k} W^{(0)T}_{:,k} X^T] \\ &= X \cdot \mathbb{E}[W^{(0)}_{:,k} W^{(0)T}_{:,k}] \cdot X^T = X \left(\frac{\sigma_w^2}{d_0} \cdot I_{d_0}\right) X^T \\ &= \frac{\sigma_w^2}{d_0} X X^T.\end{aligned}$$

Thus, if we define $\tilde{\Sigma}^{(0)} = \frac{\sigma_w^2}{d_0} X X^T$, then we have $\{X^{(0)}_{:,k}\}_{k\in[d]}$ are exactly i.i.d. from $N(\mathbf{0}_n, \tilde{\Sigma}^{(0)})$.

Induction step. The proof of the induction step is a little bit more complex than that of Proposition C.1 (NNGP correspondence for vanilla GCN). Suppose that $\{X^{(l)}_{:,k}\}_{k\in[d]}$ converge to i.i.d. n-dimensional Gaussian random variables $N(0, \tilde{\Sigma}^{(l)})$ in distribution as $d \to \infty$, then we look at the $(l+1)$-th layer. We define the characteristic function of a random variable Z by

$$\varphi_Z(\mathbf{v}) = \mathbb{E}\left[\exp(i \cdot \mathbf{v}^T Z)\right], \quad \text{for any } \mathbf{v} \in \mathbb{R}^{n\times 1},$$

and denote the characteristic function of an $n \times m$ random variable $(Z_1, Z_2, \ldots, Z_m)$ by

$$\varphi_{(Z_1,\ldots,Z_m)}(\mathbf{v}_1, \ldots, \mathbf{v}_m) = \mathbb{E}\left[\exp\left(i \cdot \sum_{k=1}^m \mathbf{v}_k^T Z_k\right)\right] \quad \text{for any } \mathbf{v}_1, \ldots, \mathbf{v}_m \in \mathbb{R}^{n\times 1}.$$

In order to demonstrate that $\{X^{(l+1)}_{:,k}\}$ converge to i.i.d. n-dimensional Gaussian random variables $N(0, \tilde{\Sigma}^{(l+1)})$ in distribution as $d \to \infty$, we only need to show that

$$\varphi_{(X^{(l+1)}_{:,k})_{k\in\mathcal{K}}}((\mathbf{v}_k)_{k\in\mathcal{K}}) \to \prod_{k\in\mathcal{K}} \exp\left(-\frac{1}{2} \mathbf{v}_k^T \tilde{\Sigma}^{(l+1)} \mathbf{v}_k\right), \quad \text{for any } \mathcal{K} \subset \mathbb{N},\ \{\mathbf{v}_k\}_{k\in\mathcal{K}} \subset \mathbb{R}^{n\times 1}.$$

Recalling (44), we take the k-th channel of $X^{(l+1)}$ at initialization and get

$$\begin{aligned} X^{(l+1)}_{:,k} &= \alpha(\hat{A}X^{(l)}W^{(l+1)}_{:,k} + \mathbf{1}_n \cdot b^{(l+1)}_k) + \beta H^{(l)}_{:,k} \\ &= \alpha\hat{A}X^{(l)}W^{(l+1)}_{:,k} + \beta X^{(l)}_{:,k}, \qquad \text{for any } k \in [d]. \end{aligned}$$

Given any fixed $\mathcal{K} \subset \mathbb{N}$ and $\{\mathbf{v}_k\}_{k\in\mathcal{K}} \subset \mathbb{R}^{n\times 1}$, then we have

$$\sum_{k\in\mathcal{K}} \mathbf{v}_k^T X^{(l+1)}_{:,k} = \alpha \sum_{k\in\mathcal{K}} \mathbf{v}_k^T \hat{A}X^{(l)}W^{(l+1)}_{:,k} + \beta \sum_{k\in\mathcal{K}} \mathbf{v}_k^T X^{(l)}_{:,k}.$$

Next we get

$$\exp\left(i \cdot \sum_{k\in\mathcal{K}} \mathbf{v}_k^T X^{(l+1)}_{:,k}\right) = \prod_{k\in\mathcal{K}} \exp\left(i \cdot \alpha \mathbf{v}_k^T \hat{A}X^{(l)}W^{(l+1)}_{:,k}\right) \cdot \prod_{k\in\mathcal{K}} \exp\left(i \cdot \beta \mathbf{v}_k^T X^{(l)}_{:,k}\right).$$

Taking the conditional expectation on both sides given $H^{(l)}$, we have

$$\begin{aligned} &\mathbb{E}\left[\exp\left(\sum_{k\in\mathcal{K}} i \cdot \mathbf{v}_k^T X^{(l+1)}_{:,k}\right) \middle| X^{(l)}\right] \\ &= \prod_{k\in\mathcal{K}} \mathbb{E}\left[\exp\left(i \cdot \alpha \mathbf{v}_k^T \hat{A}X^{(l)}W^{(l+1)}_{:,k}\right) \middle| X^{(l)}\right] \cdot \prod_{k\in\mathcal{K}} \exp\left(i \cdot \beta \mathbf{v}_k^T X^{(l)}_{:,k}\right) \qquad (46) \\ &= \prod_{k\in\mathcal{K}} \exp\left(\frac{-\sigma_w^2\alpha^2}{2d} \mathbf{v}_k^T \hat{A}X^{(l)}X^{(l)T}\hat{A}\mathbf{v}_k\right) \cdot \prod_{k\in\mathcal{K}} \exp\left(i \cdot \beta \mathbf{v}_k^T X^{(l)}_{:,k}\right). \end{aligned}$$

By the induction hypothesis, $\{X^{(l)}_{:,k}\}_{k=1}^d$ converge to i.i.d. Gaussian random variables $N(0, \tilde{\Sigma}^{(l)})$ in distribution as $d \to \infty$. By the law of large numbers, as $d \to \infty$, the first term in (46)

$$\prod_{k\in\mathcal{K}} \exp\left(\frac{-\sigma_w^2\alpha^2}{2d} \mathbf{v}_k^T \hat{A}X^{(l)}X^{(l)T}\hat{A}\mathbf{v}_k\right) \to \prod_{k\in\mathcal{K}} \exp\left(\frac{-\sigma_w^2\alpha^2}{2} \mathbf{v}_k^T \hat{A}\Sigma^{(l)}\hat{A}\mathbf{v}_k\right), \ \mathbb{P}\text{-almost surely.}$$

Thus, if we set $\tilde{\Sigma}^{(l+1)} = \sigma_w^2\alpha^2\hat{A}\tilde{\Sigma}^{(l)}\hat{A} + \beta^2\tilde{\Sigma}^{(l)}$, as $d \to \infty$, we have

$$\begin{aligned} \varphi_{(X^{(l+1)}_{:,k})_{k\in\mathcal{K}}}((\mathbf{v}_k)_{k\in\mathcal{K}}) &= \mathbb{E}\left[\exp\left(\sum_{k\in\mathcal{K}} i \cdot \mathbf{v}_k^T X^{(l+1)}_{:,k}\right)\right] \\ \to \prod_{k\in\mathcal{K}} &\exp\left(-\frac{\sigma_w^2\alpha^2}{2} \mathbf{v}_k^T \hat{A}\tilde{\Sigma}^{(l)}\hat{A}\mathbf{v}_k\right) \cdot \prod_{k\in\mathcal{K}} \exp\left(-\frac{\beta^2}{2} \mathbf{v}_k^T \tilde{\Sigma}^{(l)}\mathbf{v}_k\right) \\ = \prod_{k\in\mathcal{K}} &\exp\left(-\frac{1}{2} \mathbf{v}_k^T \tilde{\Sigma}^{(l+1)}\mathbf{v}_k\right). \end{aligned}$$

Therefore, $\{X^{(l+1)}_{:,k}\}_{k=1}^d$ converge to i.i.d. Gaussian random variables $N(0, \tilde{\Sigma}^{(l+1)})$ in distribution as $d \to \infty$.

Conclusion. By the principle of mathematical induction, we have proven the **first part** of this proposition.

Next, we will show the **second part**. For an L-layer ResGCN, under the initialzation in Assumption 3.1, the output matrix $H^{(\text{out},L)} \in \mathbb{R}^{n\times C}$ is expressed as

$$H^{(\text{out},L)} = X^{(L)}W^{(\text{out},L)} + \mathbf{1}_n \cdot b^{(\text{out},L)} = X^{(L)}W^{(\text{out},L)},$$

Similar to the proof in Proposition C.1 (NNGP correspondence for vanilla GCN), we vectorize the both sides and get

$$\text{vec}(H^{(\text{out},L)}) = \text{vec}(X^{(L)}W^{(\text{out},L)}) = \sum_{k=1}^d \text{vec}\left(\underbrace{X^{(L)}_{:,k}}_{n\times 1} \cdot \underbrace{W^{(\text{out},L)}_{k,:}}_{1\times d}\right). \qquad (47)$$

For brevity, we define

$$\omega_{kk'}^{(\mathrm{out},L)} := \sqrt{d}\cdot W_{kk'}^{(\mathrm{out},L)}, \quad \text{for all } k\in[d] \text{ and } k'\in[C],$$

and

$$Z_k^{(\mathrm{out},L)} := \mathrm{vec}\left(X_{:,k}^{(L)}\cdot \omega_{k,:}^{(\mathrm{out},L)}\right), \quad \text{for all } k\in[d].$$

Then we get that $\{\omega_{kk'}^{(\mathrm{out},L)}\}_{k\in[d],k'\in[C]}$ are i.i.d. from $N(0,\sigma_w^2)$ and

$$\text{RHS of (47)} = \frac{1}{\sqrt{d}}\sum_{k=1}^{d} Z_k^{(\mathrm{out},L)}. \tag{48}$$

According to the first part of this proposition, as $d\to\infty$, $\{X_{:,k}^{(L)}\}_{k\in[d]}$ converge to i.i.d. n-dimensional random vectors in distribution, thus $\{Z_k^{(\mathrm{out},L)}\}_{k\in[d]}$ converge to i.i.d. random vectors in distribution. Moreover, in this limiting case, by taking the expectation of (47), we have

$$\mathbb{E}[Z_1^{(\mathrm{out},L)}] = \mathrm{vec}\left(\mathbb{E}[X_{:,1}^{(L)}]\cdot\mathbb{E}[\omega_{1,:}^{(\mathrm{out},L)}]\right) = \mathrm{vec}\left(\mathbf{0}_{n\times 1}\cdot\mathbf{0}_{1\times C}\right) = \mathbf{0}_{nC}.$$

Calculating the covariance matrix of (47), we have

$$\begin{aligned}
\mathrm{Cov}[Z_1^{(\mathrm{out},L)}, Z_1^{(\mathrm{out},L)}] &= \mathbb{E}[Z_1^{(\mathrm{out},L)}\cdot Z_1^{(\mathrm{out},L)T}]\\
&= \mathbb{E}\left[\mathrm{vec}\left(X_{:,1}^{(L)}\cdot\omega_{1,:}^{(\mathrm{out},L)}\right)\cdot\mathrm{vec}\left(X_{:,1}^{(L)}\cdot\omega_{1,:}^{(\mathrm{out},L)}\right)^T\right]\\
&= \mathbb{E}\left[(\omega_{1,:}^{(\mathrm{out},L)T}\otimes X_{:,1}^{(L)})\cdot(\omega_{1,:}^{(\mathrm{out},L)}\otimes X_{:,1}^{(L)T})\right]\\
&= \mathbb{E}\left[\omega_{1,:}^{(\mathrm{out},L)T}\omega_{1,:}^{(\mathrm{out},L)}\otimes X_{:,1}^{(L)}X_{:,1}^{(L)T}\right]\\
&= \mathbb{E}\left[\omega_{1,:}^{(\mathrm{out},L)T}\omega_{1,:}^{(\mathrm{out},L)}\right]\otimes\mathbb{E}\left[X_{:,1}^{(L)}X_{:,1}^{(L)T}\right]\\
&= \sigma_w^2 I_C\otimes\tilde{\Sigma}^{(L)}\\
&= I_C\otimes\sigma_w^2\tilde{\Sigma}^{(L)}.
\end{aligned}$$

By multivariate central limit theorem, $\frac{1}{\sqrt{d}}\sum_{k=1}^{d} Z_k^{(\mathrm{out},L)}$ converges to a Gaussian random variable $N(\mathbf{0}_{nC}, I_C\otimes\sigma_w^2\tilde{\Sigma}^{(L)})$ in distribution as $d\to\infty$. Recalling (47) and (48), all the channels of $H^{(\mathrm{out},L)}$ converge to i.i.d. Gaussian random variables $N(\mathbf{0}_n,\tilde{\Sigma}^{(\mathrm{out},L)})$, where $\tilde{\Sigma}^{(\mathrm{out},L)} = \sigma_w^2\tilde{\Sigma}^{(L)}$.

Now we have complete the proof of the second part. □

D.2. Proof of Theorem 4.3 (signal propagation on linear ResGCN)

Theorem D.2. *Suppose that there exists an eigenvector u of $\hat{A}$ corresponding to the eigenvalue 1, such that the input feature $X\in\mathbb{R}^{n\times d_0}$ satisfies $X^Tu\neq\mathbf{0}_{d_0\times 1}$. Under the initialization in Assumption 3.1 and the NNGP correspondence approximation for linear ResGCN, if $\alpha^2\sigma_w^2+\beta^2>1$, then we have*

1. $\lim_{L\to\infty}\mathbb{E}_{H\sim N(\mathbf{0}_n,\tilde{\Sigma}^{(\mathrm{out},L)})}[\|H\|_{\mathrm{F}}^2/\|X\|_{\mathrm{F}}^2] = +\infty$.
2. $\lim_{L\to\infty}\mathbb{E}_{H\sim N(\mathbf{0}_n,\tilde{\Sigma}^{(\mathrm{out},L)})}[\mathrm{Dir}(H)/\|H\|_{\mathrm{F}}^2] = 0$.

Proof of part 1 in Theorem 4.3. For any positive semi-definite matrix $\Sigma\in\mathcal{S}$, we have

$$\mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\|h\|^2] = \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\mathrm{tr}(h^Th)] = \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\mathrm{tr}(hh^T)] = \mathrm{tr}(\Sigma).$$

Recalling the NNGP correspondence formula for linear ResGCN (42) in Proposition D.1, we have

$$\begin{aligned}
\tilde{\Sigma}^{(0)} &= \frac{\sigma_w^2}{d_0}XX^T,\\
\tilde{\Sigma}^{(l+1)} &= \sigma_w^2\alpha^2\hat{A}\tilde{\Sigma}^{(l)}\hat{A}+\beta^2\tilde{\Sigma}^{(l)}.
\end{aligned}\tag{49}$$

By Proposition B.1, we can assume that $A = U\Lambda U^T$, where $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ with $1 = \lambda_1 \geq \cdots \geq \lambda_n > -1$ and $U \in \mathbb{R}^{n\times n}$ is an orthogonal matrix, i.e., $UU^T = U^TU = I_n$. Then from (49), we get

$$\begin{aligned} U^T\tilde{\Sigma}^{(l+1)}U &= \sigma_w^2\alpha^2 \cdot U^T\hat{A}\tilde{\Sigma}^{(l)}\hat{A}U + \beta^2 \cdot U^T\tilde{\Sigma}^{(l)}U \\ &= \sigma_w^2\alpha^2 \cdot \Lambda U^T\tilde{\Sigma}^{(l)}U\Lambda + \beta^2 \cdot U^T\tilde{\Sigma}^{(l)}U. \end{aligned} \tag{50}$$

So for any $i \in [n]$ and $l \in \mathbb{N}$, we have

$$\begin{aligned} (U^T\tilde{\Sigma}^{(l+1)}U)_{ii} &= \sigma_w^2\alpha^2 \cdot \lambda_i(U^T\tilde{\Sigma}^{(l)}U)_{ii}\lambda_i + \beta^2(U^T\tilde{\Sigma}^{(l)}U)_{ii} \\ &= (\alpha^2\lambda_i^2\sigma_w^2 + \beta^2) \cdot (U^T\tilde{\Sigma}^{(l)}U)_{ii}. \end{aligned}$$

Thus, for any $i \in [n]$ and $L \in \mathbb{N}$, we have

$$(U^T\tilde{\Sigma}^{(L)}U)_{ii} = (\alpha^2\lambda_i^2\sigma_w^2 + \beta^2)^L \cdot (U^T\tilde{\Sigma}^{(0)}U)_{ii}.$$

According to the assumption on input feature X, there exists an eigenvector u of $\hat{A}$ corresponding to the eigenvalue 1, such that $X^Tu \neq \mathbf{0}_{d_0\times 1}$. Suppose that $u_1, u_2, \ldots, u_n \in \mathbb{R}^{n\times 1}$ are the columns of U, then there exists $i \in [M]$ such that $X^Tu_i \neq 0$. Otherwise, suppose that $u = \sum_{j=1}^M c_ju_j$ and $X^Tu_j = 0$ for any $j \in [M]$, then $X^Tu = \sum_{j=1}^M c_jX^Tu_j = 0$. Contradiction!

Without loss of generality, we suppose that $Au_1 = u_1$ and $X^Tu_1 \neq \mathbf{0}_{d\times 1}$. Then we have

$$(U^T\tilde{\Sigma}^{(0)}U)_{11} = \frac{\sigma_w^2}{d_0} \cdot u_1^TXX^Tu_1 = \frac{\sigma_w^2}{d_0} \cdot \|X^Tu_1\|^2 > 0.$$

It results in

$$\operatorname{tr}(\tilde{\Sigma}^{(L)}) = \operatorname{tr}(U^T\tilde{\Sigma}^{(L)}U) \geq (U^T\tilde{\Sigma}^{(L)}U)_{11} = (\alpha^2\sigma_w^2 + \beta^2)^L \cdot \frac{\sigma_w^2}{d_0}\|X^Tu_1\|^2.$$

By (42) in Proposition D.1, we have

$$\operatorname{tr}(\tilde{\Sigma}^{(\mathrm{out},L)}) = \sigma_w^2 \operatorname{tr}(\tilde{\Sigma}^{(L)}) \geq (\alpha^2\sigma_w^2 + \beta^2)^L \cdot \frac{\sigma_w^4}{d_0}\|X^Tu_1\|^2. \tag{51}$$

Therefore, if $\alpha^2\sigma_w^2 + \beta^2 > 1$, we have

$$\begin{aligned} \lim_{L\to\infty} \mathbb{E}_{H\sim N(\mathbf{0}_n,\tilde{\Sigma}^{(\mathrm{out},L)})}[\|H\|_{\mathrm{F}}^2/\|X\|_{\mathrm{F}}^2] &= \frac{C}{\|X\|_{\mathrm{F}}^2} \lim_{L\to\infty} \mathbb{E}_{h\sim N(\mathbf{0}_n,\tilde{\Sigma}^{(\mathrm{out},L)})}[\|h\|^2] \\ &= \frac{C}{\|X\|_F^2} \lim_{L\to\infty} \operatorname{tr}(\tilde{\Sigma}^{(\mathrm{out},L)}) = +\infty. \end{aligned}$$

□

Proof of part 2 *in Theorem 4.3*. For any positive semi-definite matrix $\Sigma \in \mathcal{S}$, we have

$$\mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\operatorname{Dir}(h)] = \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\operatorname{tr}(h^T\hat{L}h)] = \mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\operatorname{tr}(\hat{L}hh^T)] = \operatorname{tr}(\hat{L}\Sigma).$$

So when we want to study $\mathbb{E}_{h\sim N(\mathbf{0}_n,\Sigma)}[\operatorname{Dir}(h)]$, we only need to look at $\operatorname{tr}(\hat{L}\Sigma)$ in the following of the proof.

Since $\hat{A}\hat{L} = \hat{A}(I_n - \hat{A}) = \hat{A} - \hat{A}^2 = (I_n - \hat{A})\hat{A} = \hat{L}\hat{A}$, we multiply $\hat{L}$ on both sides of the second equation in (49) and get

$$\begin{aligned} \hat{L}\tilde{\Sigma}^{(l+1)} &= \sigma_w^2\alpha^2 \cdot \hat{L}\hat{A}\tilde{\Sigma}^{(l)}\hat{A} + \beta^2\hat{L}\Sigma^{(l)} \\ &= \sigma_w^2\alpha^2 \cdot \hat{A}\hat{L}\tilde{\Sigma}^{(l)}\hat{A} + \beta^2\hat{L}\Sigma^{(l)}. \end{aligned}$$

Then for any $i \in [n]$ and $l \in \mathbb{N}$, we have

$$\begin{aligned} (U^T\hat{L}\tilde{\Sigma}^{(l)}U)_{ii} &= \sigma_w^2\alpha^2 \cdot \lambda_i(U^T\hat{L}\tilde{\Sigma}^{(0)}U)_{ii}\lambda_i + \beta^2 \cdot (U^T\hat{L}\tilde{\Sigma}^{(0)}U)_{ii} \\ &= (\alpha^2\lambda_i^2\sigma_w^2 + \beta^2) \cdot (U^T\hat{L}\tilde{\Sigma}^{(0)}U)_{ii}. \end{aligned}$$

Thus, for any $i \in [n]$ and $L \in \mathbb{N}$, we have

$$(U^T \hat{L} \tilde{\Sigma}^{(L)} U)_{ii} = (\alpha^2 \sigma_w^2 \lambda_i^2 + \beta^2)^L \cdot (U^T \hat{L} \tilde{\Sigma}^{(0)} U)_{ii} \tag{52}$$

Since $U^T \hat{L} U = U^T (I_n - \hat{A}) U = I_n - \Lambda$, we get

$$U^T \hat{L} \tilde{\Sigma}^{(0)} U = (I_n - \Lambda) U^T \tilde{\Sigma}^{(0)} U$$

We denote

$$r_i = (U^T \tilde{\Sigma}^{(0)} U)_{ii}, \quad \text{for any } i \in [n].$$

Then by (52), we have

$$(U^T \hat{L} \tilde{\Sigma}^{(L)} U)_{ii} = (\alpha^2 \sigma_w^2 \lambda_i^2 + \beta^2)^L \cdot (1 - \lambda_i) r_i,$$

From Proposition B.1, we have

$$\begin{aligned} (U^T \hat{L} \tilde{\Sigma}^{(L)} U)_{ii} &\leq (\alpha^2 \sigma_w^2 \lambda^2 + \beta^2)^L \cdot (1 - \lambda_i) r_i, \quad \text{if } \lambda_i \in (-1, 1); \\ (U^T \hat{L} \tilde{\Sigma}^{(L)} U)_{ii} = 0 &= (\alpha^2 \sigma_w^2 \lambda^2 + \beta^2)^L \cdot (1 - \lambda_i) r_i, \quad \text{if } \lambda_i = 1, \end{aligned}$$

where $\lambda = \max_{\lambda_i \neq 1} |\lambda_i| \in [0, 1)$. Thus, we get

$$\operatorname{tr}(\hat{L} \tilde{\Sigma}^{(\text{out}, L)}) = \sigma_w^2 \operatorname{tr}(\hat{L} \tilde{\Sigma}^{(L)}) = \operatorname{tr}(U^T \hat{L} \tilde{\Sigma}^{(L)} U) \leq \sigma_w^2 (\alpha^2 \sigma_w^2 \lambda^2 + \beta^2)^L \cdot \sum_{i=1}^n (1 - \lambda_i) r_i.$$

We conclude that

$$\begin{aligned} \mathbb{E}_{H \sim N(\mathbf{0}_n, \tilde{\Sigma}^{(\text{out}, L)})}[\operatorname{Dir}(H)] &= C \cdot \mathbb{E}_{h \sim N(\mathbf{0}_n, \tilde{\Sigma}^{(L)})}[\operatorname{Dir}(h)] = C \cdot \operatorname{tr}(\hat{L} \tilde{\Sigma}^{(L)}) \\ &\leq C \sigma_w^2 (\alpha^2 \sigma_w^2 \lambda^2 + \beta^2)^L \cdot \sum_{i=1}^n (1 - \lambda_i) r_i. \end{aligned}$$

Then we have

$$\frac{\mathbb{E}_{H \sim N(\mathbf{0}_n, \tilde{\Sigma}^{(\text{out}, L)})}[\operatorname{Dir}(H)]}{(\alpha^2 \sigma_w^2 + \beta^2)^L} \leq \left(C \sigma_w^2 \sum_{i=1}^n (1 - \lambda_i) r_i \right) \cdot \left(\frac{\alpha^2 \sigma_w^2 \lambda^2 + \beta^2}{\alpha^2 \sigma_w^2 + \beta^2} \right)^L.$$

Since $\alpha^2 + \sigma_w^2 + \beta^2 > 1$ and $\alpha \neq 0$ as assumed in the statement of this theorem, we have $(\alpha^2 \sigma_w^2 \lambda^2 + \beta^2)/(\alpha^2 \sigma_w^2 + \beta^2) \in [0, 1)$. So we get that

$$\lim_{L \to \infty} \frac{\mathbb{E}_{H \sim N(\mathbf{0}_n, \tilde{\Sigma}^{(\text{out}, L)})}[\operatorname{Dir}(H)]}{(\alpha^2 \sigma_w^2 + \beta^2)^L} = 0. \tag{53}$$

Recalling (51) in part 1 of the proof, if we define

$$\delta_0 = \frac{\sigma_w^4}{d_0} \|X^T u_1\| \quad \text{and} \quad K = \alpha^2 \sigma_w^2 + \beta^2,$$

then given any $L \in \mathbb{N}$, we have

$$\frac{1}{K^L} \cdot \operatorname{tr}(\tilde{\Sigma}^{(\text{out}, L)}) \geq \delta_0 > 0. \tag{54}$$

Similar to the proof of part 2 in Theorem C.7, we have

$$\begin{aligned} &\mathbb{E}_{H \sim N(\mathbf{0}_n, \tilde{\Sigma}^{(\text{out}, L)})} \left[\frac{\operatorname{Dir}(H)}{\|H\|_{\text{F}}^2} \right] \\ &= \mathbb{E}_{H \sim N(\mathbf{0}_n, \tilde{\Sigma}^{(\text{out}, L)})} \left[\frac{\operatorname{Dir}(H)}{\|H\|_{\text{F}}^2} \mathbb{1}_{\{\|H\|_{\text{F}}^2 > \epsilon K^L\}} \right] + \mathbb{E}_{H \sim N(\mathbf{0}_n, \tilde{\Sigma}^{(\text{out}, L)})} \left[\frac{\operatorname{Dir}(H)}{\|H\|_{\text{F}}^2} \mathbb{1}_{\{\|H\|_{\text{F}}^2 \leq \epsilon K^L\}} \right] \\ &\leq \frac{\mathbb{E}_{H \sim N(\mathbf{0}_n, \tilde{\Sigma}^{(\text{out}, L)})}[\operatorname{Dir}(H)]}{\epsilon K^L} + 2 \cdot \mathbb{P}_{H \sim N(\mathbf{0}_n, \tilde{\Sigma}^{(\text{out}, L)})}[\|H\|_{\text{F}}^2 \leq \epsilon K^L]. \end{aligned} \tag{55}$$

For any $L \geq 1$, there exists $i \in [n]$, such that $\Sigma_{ii}^{(\mathrm{out},L)} \geq \mathrm{tr}(\Sigma^{(\mathrm{out},L)})/n$. For any $n \times C$ random matrix $H \sim N(\mathbf{0}_n, \Sigma^{(\mathrm{out},L)})$, it holds that $H_{i,1} \sim N(0, \Sigma_{ii}^{(\mathrm{out},L)})$. By (54), we have

$$\begin{aligned}
&\mathbb{P}_{H\sim N(\mathbf{0}_n,\tilde{\Sigma}^{(\mathrm{out},L)})}\left[\|H\|_{\mathrm{F}}^2 \leq \epsilon K^L\right] \leq \mathbb{P}_{H\sim N(\mathbf{0}_n,\tilde{\Sigma}^{(\mathrm{out},L)})}\left[H_{i,1}^2 \leq \epsilon K^L\right] \\
&= \mathbb{P}_{Z\sim N(0,1)}\left[Z^2 \leq \frac{\epsilon K^L}{\Sigma_{ii}^{(\mathrm{out},L)}}\right] \leq \mathbb{P}_{Z\sim N(0,1)}\left[Z^2 \leq \frac{\epsilon n K^L}{\mathrm{tr}(\Sigma^{(\mathrm{out},L)})}\right] \\
&\leq \mathbb{P}_{Z\sim N(0,1)}\left[Z^2 \leq \frac{\epsilon n}{\delta_0}\right] = 2\Phi\left(\sqrt{\frac{\epsilon n}{\delta_0}}\right) - 1,
\end{aligned} \tag{56}$$

where $\Phi(x) = \mathbb{P}_{Z\sim N(0,1)}[Z \leq x]$ denotes the cumulative distribution function of the standard normal distribution $N(0,1)$. Combining (55) and (56), we get

$$\mathbb{E}_{H\sim N(\mathbf{0}_n,\tilde{\Sigma}^{(\mathrm{out},L)})}\left[\frac{\mathrm{Dir}(H)}{\|H\|_{\mathrm{F}}^2}\right] \leq \frac{1}{\epsilon K^L} \cdot \mathbb{E}_{H\sim N(\mathbf{0}_n,\tilde{\Sigma}^{(\mathrm{out},L)})}[\mathrm{Dir}(H)] + 4\Phi\left(\sqrt{\frac{\epsilon n}{\delta_0}}\right) - 2.$$

By (53) , we let $L \to \infty$ and get

$$\begin{aligned}
&\limsup_{L\to\infty} \mathbb{E}_{H\sim N(\mathbf{0}_n,\tilde{\Sigma}^{(\mathrm{out},L)})}\left[\frac{\mathrm{Dir}(H)}{\|H\|_{\mathrm{F}}^2}\right] \\
&\leq \frac{1}{\epsilon K^L} \cdot \limsup_{L\to\infty} \mathbb{E}_{H\sim N(\mathbf{0}_n,\tilde{\Sigma}^{(\mathrm{out},L)})}[\mathrm{Dir}(H)] + 4\Phi\left(\sqrt{\frac{\epsilon n}{\delta_0}}\right) - 2 \\
&= \frac{1}{\epsilon} \cdot 0 + 4\Phi\left(\sqrt{\frac{\epsilon n}{\delta_0}}\right) - 2 = 4\Phi\left(\sqrt{\frac{\epsilon n}{\delta_0}}\right) - 2.
\end{aligned} \tag{57}$$

Notice that the left hand side of (57) is independent of the choice of ϵ. Since Φ is a continuous map, we let $\epsilon \to 0^+$ and get

$$\limsup_{L\to\infty} \mathbb{E}_{H\sim N(\mathbf{0}_n,\tilde{\Sigma}^{(\mathrm{out},L)})}\left[\frac{\mathrm{Dir}(H)}{\|H\|_{\mathrm{F}}^2}\right] \leq 4\Phi(0) - 2 = 0.$$

Therefore, we have

$$\lim_{L\to\infty} \mathbb{E}_{H\sim N(\mathbf{0}_n,\tilde{\Sigma}^{(\mathrm{out},L)})}\left[\frac{\mathrm{Dir}(H)}{\|H\|_{\mathrm{F}}^2}\right] = 0.$$

□

E. Best reported depths of existing over-smoothing-related approaches on OGBN-Arxiv

Table 2. Summary of depths with optimal test accuracies of existing over-smoothing-related approaches on OGNB-Arxiv. The corresponding optimal test accuracies are also indicated in parentheses.

Models	Depth (test accuracy)
GCN(Kipf & Welling, 2017)	2 (69.53)
DropEdge(Rong et al., 2020)	2 (68.67)
PairNorm(Zhao & Akoglu, 2020)	2 (65.74)
DropNode(Huang et al., 2020)	16 (67.17)
MeanNorm(Yang et al., 2020)	16 (70.40)
GroupNorm(Zhou et al., 2020)	16 (70.50)
NodeNorm(Zhou et al., 2021a)	16 (70.75)
GCNII(Chen et al., 2020b)	16 (72.61)
GPRGNN(Chien et al., 2021)	16 (70.30)
DAGNN(Liu et al., 2020)	16 (71.82)
EGNN(Zhou et al., 2021b)	32 (72.7)
JKNet(Xu et al., 2018)	16 (66.41)
APPNP(Gasteiger et al., 2019)	16 (66.95)
ReZeroGCN(Ours)	64 (72.97)

In Table 2, we summarize the best depths of existing approaches aiming to tackle the over-smoothing issue. The best depth refers to the depth that achieves the highest test accuracy on OGBN-Arxiv. For approaches that utilize random dropping or normalization techniques, we all employ vanilla GCNs as the underlying model. Most of the results are directly cited from (Chen et al., 2022b), while the result of EGNN is cited from the original paper (Zhou et al., 2021b). We see that the best performance in most of these studies is still achieved with less than 20 layers, suggesting that the curse of depth continues to constrain the potential of GCNs.

F. SPoGInit algorithm details

In lines 1-2, we initialize the weight parameters by following Xavier initialization and set the initial scale $\gamma^{(l)}(0)$ to be 1 for every layer l. Moreover, we initialize $\theta(\gamma(0))$ with the settings $\theta(\gamma) = \{W^{(l)}\}_{l=1}^{L} = \{\gamma^{(l)}\hat{W}^{(l)}\}_{l=1}^{L}$ in Section 5.1. We iteratively update $\theta(\gamma)$ as follows.

In lines 4-5, at each interaction, we sample random node features from a standard Gaussian distribution and node labels from a discrete uniform distribution. Borrowing the idea from MetaInit (Dauphin & Schoenholz, 2019), this approach is to enhance data independence and solve over-fitting problems. In line 6, we calculate the objective function $F(\theta(\gamma(t)))$ as defined in Section 5.1:

$$F(\theta(\gamma)) := w_1 \underbrace{\left[\frac{\|H^{(1)}(\theta(\gamma))\|_{\mathrm{F}}}{\|H^{(L-1)}(\theta(\gamma))\|_{\mathrm{F}}} - 1\right]^2}_{(a)} + w_2 \underbrace{\left[\frac{\|g^{(2)}(\theta(\gamma))\|_{\mathrm{F}}}{\|g^{(L-1)}(\theta(\gamma))\|_{\mathrm{F}}} - 1\right]^2}_{(b)} - w_3 \underbrace{\frac{\mathrm{Dir}(H^{(L)}(\theta(\gamma)))}{\|H^{(L)}(\theta(\gamma)\|_{\mathrm{F}}^2}}_{(c)}.$$

To be more specific, given the model parameters $\theta(\gamma)$, random features and the normalized adjacency matrix $\hat{A}$, we can get the embeddings $H^{(1)}(\theta(\gamma))$, $H^{(L-1)}(\theta(\gamma))$, and $H^{(L)}(\theta(\gamma))$, which can be used to calculate term (a) and term (c). Combined with the labels of the training nodes, we can get the training loss function and the gradients $g^{(2)}(\theta(\gamma))$ and $g^{(L-1)}(\theta(\gamma))$, which can be used to calculate term (b). Here, $g^{(l)}(\theta(\gamma)) = \partial\ell/\partial W^{(l)}$ for each $l \in [L]$ as claimed in Section 5.1.

Then, in lines 7-12, we update the weight parameters $\theta(\gamma(t))$ by optimizing the objective function through the projected gradient descent method to the scales $\{\gamma^{(l)}(t)\}_{l=1}^{L}$ for each layer. We adopt the projected gradient descent method to ensure the scales $\{\gamma^{(l)}(t)\}_{l=1}^{L}$ remain positive.

Algorithm 1 *SPoGInit*: searching for weight initialization with better Signal Propagation on Graph

Input: normalized adjacency matrix $\hat{A}$, input dimension d_0, number of labels C, network depth L, hidden dimension d, learning rate η, total iterations T, metric weights w_1, w_2, w_3.

```
 1: initialize $\gamma^{(l)}(0) = 1$ and generate $\{\hat{W}^{(l)}\}_{l=1}^{L}$ with $\hat{W}_{k'k}^{(l)} \overset{\text{iid}}{\sim} \text{Uniform}(-\sqrt{\frac{6}{d_{l-1}+d_l}}, \sqrt{\frac{6}{d_{l-1}+d_l}})$.
 2: initialize $\theta(\gamma(0)) \triangleq \{W^{(l)}(0)\}_{l=1}^{L}$ by $W^{(l)}(0) \leftarrow \gamma^{(l)}(0) \cdot \hat{W}^{(l)}$.
 3: for $t = 0, 1, \cdots, T-1$ do
 4:     generate input $X(t) \in \mathbb{R}^{n \times d_0}$ with $X(t)_{ik} \overset{\text{iid}}{\sim} N(0,1)$.
 5:     generate label $y_i(t) \overset{\text{iid}}{\sim} \text{Uniform}\{1, 2, \ldots, C\}$ for any node $i \in [n]$.
 6:     calculate the objective function $F(\theta(\gamma(t)))$ (see Section 5.1) by $\hat{A}$, $X(t)$, $y(t)$ and $\theta(\gamma(t))$.
 7:     for layers $l = 1, 2, \ldots, L$ do
 8:         $\gamma^{(l)}(t+1) \leftarrow \gamma^{(l)}(t) - \eta\nabla_{\gamma^{(l)}} F(\theta(\gamma(t)))$.
 9:         $\gamma^{(l)}(t+1) \leftarrow \text{Proj}_{[10^{-6},\infty)}(\gamma^{(l)}(t+1))$
10:         $W^{(l)}(t+1) \leftarrow \gamma^{(l)}(t+1) \cdot \hat{W}^{(l)}$.
11:     end for
12:     $\theta(\gamma(t+1)) \triangleq \{W^{(l)}(t+1)\}_{l=1}^{L}$.
13: end for
```

Return: $\theta(\gamma(T))$.

Specifically, we explain how to compute the derivative of the objective function with respect to the scale γ. Through the chain rule, we can calculate the derivative of the objective function with respect to the scale $\gamma^{(l)}$ as follows:

$$\begin{aligned} \frac{\partial F(\theta(\gamma)}{\partial \gamma^{(l)}} &= \sum_{k'=1}^{d_{l-1}} \sum_{k=1}^{d_l} \frac{\partial F(\theta(\gamma)}{\partial W_{k'k}^{(l)}} \frac{\partial W_{k'k}^{(l)}}{\partial \gamma^{(l)}} = \sum_{k'=1}^{d_{l-1}} \sum_{k=1}^{d_l} \frac{\partial F(\theta(\gamma)}{\partial W_{k'k}^{(l)}} \hat{W}_{k'k}^{(l)} \\ &= \sum_{k'=1}^{d_{l-1}} \sum_{k=1}^{d_l} \frac{\partial F(\theta(\gamma)}{\partial W_{k'k}^{(l)}} \frac{W_{k'k}^{(l)}}{\gamma^{(l)}}. \end{aligned}$$

Additionally, we provide specific hyperparameter choices for SPoGInit. We set total iterations T as 500 and learning rate η as 0.1. Considering the sensitivity of the training process to weight gradients, we assign a higher weight to the backward propagation. Finally, we set the forward propagation and backward propagation weights as $w_1 = 1$ and $w_2 = 10$. Moreover, to balance the scale of the output diversity, we utilize the inverse of the Dirichlet energy of the input data as the weight: $w_3 = \|X\|_{\mathrm{F}}^2 / \mathrm{Dir}(X)$.

G. Supplemental experiment results

G.1. Datasets

Datasets: We focus on seven benchmark datasets for semi-supervised node classification. The small-scale datasets include Cora, Pubmed, and Citeseer (Yang et al., 2016). The large-scale datasets comprise OGBN-Arxiv, IGB-tiny19, IGB-tiny2983, and Arxiv-year. These large-scale datasets are selected from three popular publicly available graph benchmarks: Open Graph Benchmark (OGB) (Hu et al., 2020), Illinois Graph Benchmark (IGB) (Khatua et al., 2023), and Large Scale Non-Homophilous Graphs Benchmark (Lim et al., 2021). We use a standard training/validation/test split (Yang et al., 2016) for Cora, Pubmed, and Citeseer datasets. On large-scale datasets, we adopt standard training/validation/test splits. Statistics of the datasets are summarized in Table 3.

Table 3. Statistics of the seven datasets used in the experiments (Section 6 and Appendix G).

Dataset	Nodes	Features	Edges	Class	Homophily	Training/Validation/Test
Cora	2708	1433	10556	7	0.81	5.2%/18.5%/36.9%
Pubmed	19717	500	88648	3	0.80	0.3%/2.5%/5.1%
Citeseer	3327	3703	9104	6	0.74	3.6%/15.0%/30.1%
OGBN-Arxiv	169343	128	1166243	40	0.66	53.7%/17.6%/28.7%
IGB-Tiny19	100000	1024	447416	19	0.56	60%/20%/20%
IGB-Tiny2983	100000	1024	447416	235	0.47	60%/20%/20%
Arxiv-year	169343	128	1166243	5	0.22	50%/25%/25%

G.2. Experimental settings and hyperpameters

Settings for the initialization experiments of vanilla GCN.

In Figures 5, 9, and 10, the number of hidden units is set to be 64. The models are trained using the Adam optimizer with the tanh activation function. For the Adam optimizer, we set the momentum coefficients to 0.9 and 0.9995, and perform grid searches over the learning rates ranging from $10^{-3}, 10^{-4}, 5 \times 10^{-5}$, to 10^{-5}. Table 4 reports the settings for training epochs and early stopping patience with different network depths. To investigate the training degradation issue, we exclude dropout (Srivastava et al., 2014) and weight decay.

In this work, we replicate all training experiments across three random seeds. Additionally, we replicate all experiments at initialization across 20 random seeds.

Table 4. Epochs settings of Figures 5, 9, and 10.

GCN layers	Hyperparameters
4/8/16 layers	epochs; 800, patience: 200
32 layers	epochs: 1200, patience: 300
64 layers	epochs: 1500, patience: 375
128 layers	epochs: 2000, patience: 400

Settings for the experiments of skip-connection-based GCNs.

- **Model performance.** In Figures 4 and 11, we set the number of hidden units to be 64. We add batch normalization and dropout to all models to enhance the generalization performance, as they are commonly used in the training of GNNs on large-scale datasets. The epoch settings of different datasets are as follows: 1000 epochs for OGBN-Arxiv, 1500 epochs for Arxiv-year, 1000 epochs for IGB-Tiny 19, and 1000 epochs for IGB-Tiny2983. The Adam optimizer's two momentum coefficients are set to be 0.9 and 0.9995. The weight decay is fixed to 0. We replicate all training experiments across three random seeds. Additional hyperparameters are reported in Table 5.
- **Backward propagation analysis.** As discussed in Section 5.2, although the gradient norms of $W^{(l)}$ in ReZeroGCN are zero at initialization, the non-zero gradients of $\alpha^{(l)}$ (See Appendix G.5) facilitate the update of $W^{(l)}$ in the following training epochs. Therefore, we evaluate the backward propagation by the gradient norms **during early training**. For the

early training experiments in Figures 7, 14, 15, 16, and 17, we maintain most of the settings in the **model performance** experiments (including learning rates, hidden units, initializations, weight decay, and momentum coefficients in Adam). To investigate backward propagation issues, we exclude batch normalization and dropout. We replicate the early-training experiments across five random seeds.

Table 5. Hyperparameters of Figures 4 and 11

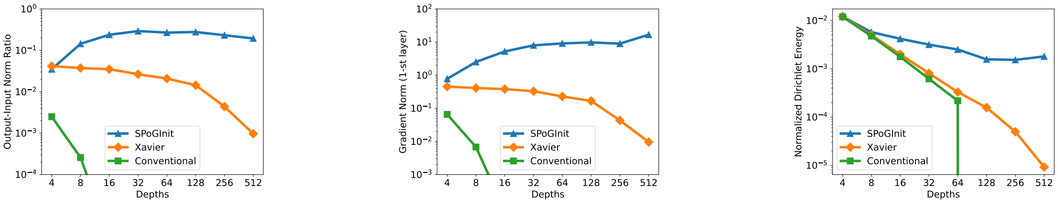
Models	Hyperparameters
JKNet	hidden units: 64, initialization: Xavier, learning rate: 0.005, dropout:0.5.
ResGCN	hidden units: 64, initialization: Conventional, learning rate: 0.005, dropout:0.6.
GCNII	α_l: 0.5, λ: 0.5, hidden units: 64, initialization: Xavier, learning rate: 0.005, dropout:0.1.
ReZeroGCN	hidden units: 64, initialization: Xavier, learning rate: 0.005, dropout:0.6.

Overall settings

All experiments on large-sized datasets, e.g., OGBN-Arxiv, are conducted on a single NVIDIA V100 32 GB GPU, while small-sized datasets experiments are completed using a single NVIDIA T4 16 GB GPU.

G.3. Additional experiments for SPoGInit

Signal propagation experiments on additional datasets

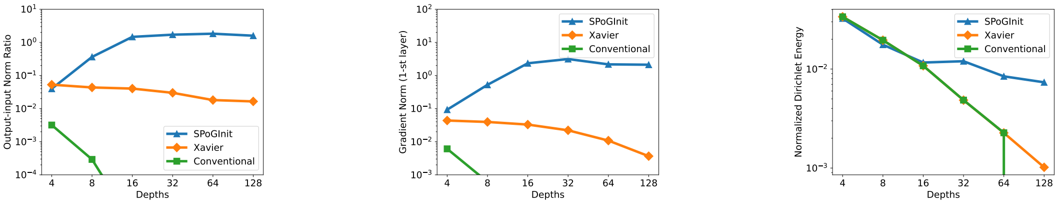


(a) Forward

(b) Backward

(c) Output diversity

Figure 9. The (a) forward metrics, (b) backward metrics, and (c) output diversity metrics of deep vanilla GCNs with different initialization methods on the Citeseer dataset.



(a) Forward

(b) Backward

(c) Output diversity

Figure 10. The (a) forward metrics, (b) backward metrics, and (c) output diversity metrics of deep vanilla GCNs with different initialization methods on the Pubmed dataset.

In Figures 9 and 10, we present the average forward propagation metrics, backward propagation metrics, and output diversity metrics of tanh-activated vanilla GCNs with various initialization methods, depths, and datasets. We replicate these experiments across 20 random seeds. Results demonstrate that deep vanilla GCNs with Xavier and Conventional initializations suffer from poor forward-backward propagation and output diversity. In contrast, SPoGInit stabilizes the forward-backward propagation and enhances the output diversity.

Performance on additional datasets

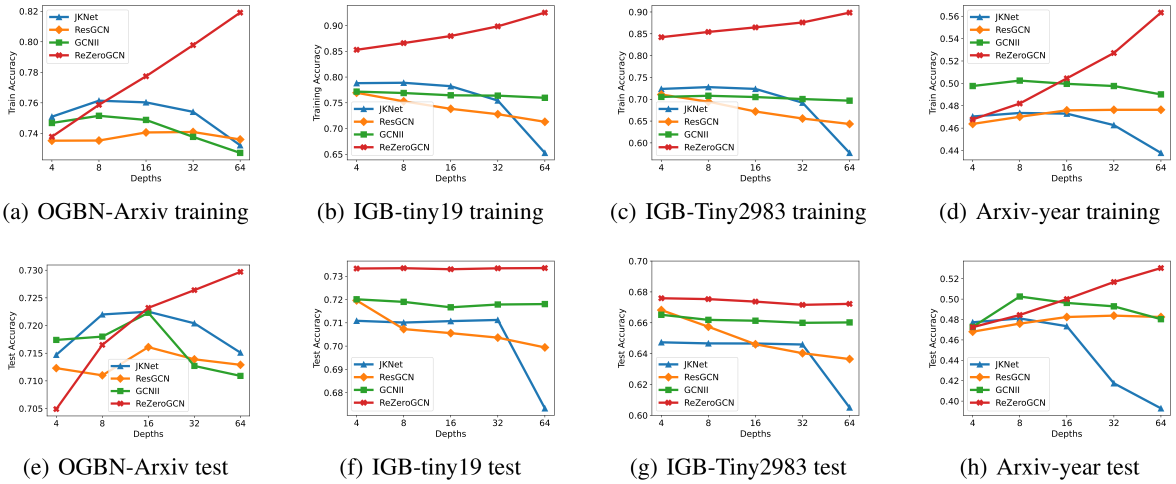
Table 6. Training and test accuracies of vanilla GCNs with different initialization methods, depths, and datasets. The abbreviation ”OOM” means out of memory.

Datasets	Init	Training accuracies for different depths						Test accuracies for different depths					
		4	8	16	32	64	128	4	8	16	32	64	128
Cora	Conventional	100	100	73.6	63.6	43.8	49.8	79.3	71.2	57.8	47.8	36.9	37.1
	Xavier	100	100	**100**	91.0	87.4	81.0	79.4	**78.4**	75.2	71.6	70.5	64.8
	SPoGInit	100	100	99.3	**100**	**92.6**	**88.1**	**79.7**	77.9	**77.0**	**74.7**	**74.0**	**72.3**
Pubmed	Conventional	100	100	88.9	73.3	75.6	60.6	76.3	72.6	67.3	68.9	62.3	49.0
	Xavier	100	100	**100**	97.8	**91.7**	74.4	**76.6**	75.9	75.9	**76.3**	**78.1**	68.7
	SPoGInit	100	100	99.4	**98.3**	89.4	**86.1**	76.3	**76.4**	**77.9**	75.9	77.2	**75.9**
Citeseer	Conventional	100	99.2	97.8	43.1	63.6	34.2	67.6	59.3	52.1	40.2	37.8	29.3
	Xavier	100	100	98.1	94.7	91.9	85.6	67.5	**67.5**	62.3	56.5	56.7	54.1
	SPoGInit	100	100	**98.3**	94.7	**93.6**	**91.4**	**67.8**	65.1	59.9	**62.2**	**57.7**	**54.9**
OGBN-Arxiv	Conventional	74.5	70.5	50.3	31.7	27.3	OOM	70.1	67.8	49.9	33.2	35.9	OOM
	Xavier	75.2	74.4	68.3	56.2	40.5	OOM	**70.4**	68.6	66.3	57.4	39.0	OOM
	SPoGInit	**75.5**	**75.1**	**70.9**	**63.5**	OOM	OOM	70.3	**69.2**	**67.7**	**63.4**	OOM	OOM

In Table 6, we present the average training and test accuracies of tanh-activated GCNs with different initialization methods, depths, and datasets. We replicate this experiment across three random seeds. The results show that vanilla GCNs with Xavier initialization and Conventional initialization suffer from performance degradation on various datasets. Our proposed SPoGInit effectively alleviates the performance degradation and outperforms the baseline initializations in deep GCNs on various datasets.

G.4. Performance of skip-connection-based GCNs

Experimental results of tanh-activated models



(a) OGBN-Arxiv training (b) IGB-tiny19 training (c) IGB-Tiny2983 training (d) Arxiv-year training

(e) OGBN-Arxiv test (f) IGB-tiny19 test (g) IGB-Tiny2983 test (h) Arxiv-year test

Figure 11. The average training accuracies (a)-(d) and test accuracies (e)-(h) of different skip-connection-based GCNs with tanh activation on various datasets.

In Figure 11, we present the average training and test accuracies of tanh-activated models with various depths. We see that ReZeroGCN stands out by achieving consistent performance gains as the depth increases. Additionally, on the OGBN-Arxiv and Arxiv-year datasets, ReZeroGCN achieves around 3% gains in test accuracy by deepening the model from 4 to 64 layers. These results demonstrate that ReZeroGCN, with both ReLU and tanh activations, successfully overcomes the curse of depths.

Optimal performance and the corresponding depths

In Tables 7 and 8, we present the optimal test accuracies and the corresponding depths for various models and datasets. These values are derived from the results presented in Figures 4 and 11. Notably, we see that ReZeroGCN outperforms all the baseline models and achieves optimal performances at the largest depths across most datasets.

Table 7. Optimal test accuracies and the corresponding depths (the numbers in parentheses) of ReLU-activated models.

Models	OGBN-Arxiv	IGB-Tiny19	IGB-Tiny2983	Arxiv-year
JKNet	72.46±0.17 (16)	70.96±0.09 (4)	64.63±0.06 (4)	51.31±0.02 (8)
ResGCN	72.46±0.12 (16)	72.08±0.15 (4)	67.06±0.10 (4)	52.98±0.09 (16)
GCNII	72.51±0.39 (8)	71.96±0.07 (4)	66.12±0.10 (4)	52.39±0.23 (8)
ReZeroGCN	**72.76±0.35** (64)	**73.36±0.05** (64)	**67.59±0.05** (4)	**54.22±0.14** (64)

Table 8. Optimal test accuracies and the corresponding depths (the numbers in parentheses) of tanh-activated models.

Models	OGBN-Arxiv	IGB-Tiny19	IGB-Tiny2983	Arxiv-year
JKNet	72.25±0.28 (16)	71.12±0.03 (32)	64.73±0.02 (4)	48.11±0.10 (8)
ResGCN	71.61±0.09 (16)	71.96±0.04 (4)	66.82±0.22 (4)	48.38±0.06 (32)
GCNII	72.23±0.26 (16)	72.01±0.12 (4)	66.52±0.07 (4)	50.25±0.06 (8)
ReZeroGCN	**72.97±0.16** (64)	**73.35±0.03** (8)	**67.59±0.06** (4)	**53.04±0.14** (64)

G.5. Backward metrics of baseline GCNs with skip connections

Skip connections significantly change the back-propagation computation. Therefore, in this subsection, we evaluate backward propagation by measuring the gradient norms of four representative layers in an L-layer model: the first, the

$L/4$-th, the $L/2$-th, and the $3L/4$-th layers. For GCNII, we evaluate its backward propagation metric by the gradient norms of $W_1^{(1)}$ (see Equation 2) in these layers. Additional settings can be seen in Appendix G.2.

Gradient norms of $\alpha^{(l)}$ in ReZeroGCN at initialization

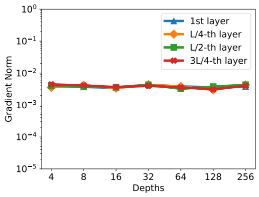
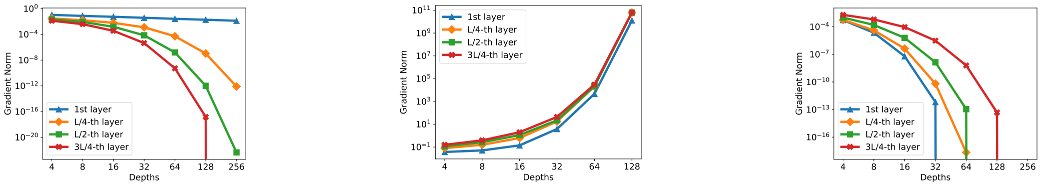


Figure 12. The gradient norms of the $\alpha^{(l)}$ in ReLU-activated ReZeroGCNs with various depths and layers at initialization on the Cora dataset. We replicate this experiment across 50 random seeds.

In Figure 12, we present the average gradient norms of $\alpha^{(l)}$ in ReLU-activated ReZeroGCNs at initialization. The results demonstrate that $\alpha^{(l)}$ in ReZeroGCNs exhibit non-vanishing and stable gradient norms at initialization in various depths and layers.

Backward metrics of baseline GCNs with skip connections at initialization



(a) JKNet

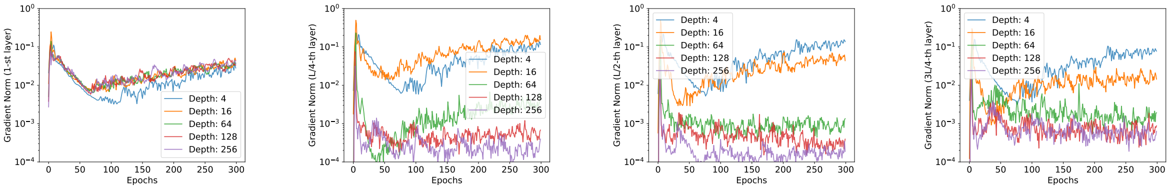
(b) ResGCN

(c) GCNII

Figure 13. The backward metrics of four layers in ReLU-activated baseline models with various depths at initialization on the Cora dataset. We replicate this experiment across 20 random seeds.

In Figure 13, we present the average backward metrics of baselines with different depths at initialization. Results demonstrate that JKNet and GCNII suffer from serious gradient vanishing problems at initialization, while the gradient norms of ResGCN explode at initialization.

Backward metrics of GCNs with skip connections during early training



(a) 1-st layer

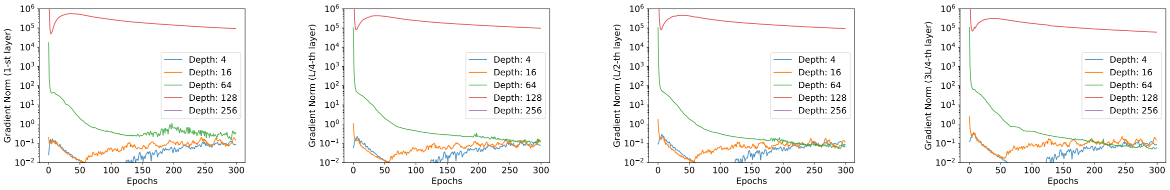
(b) $L/4$-th layer

(c) $L/2$-th layer

(d) $3L/4$-th layer

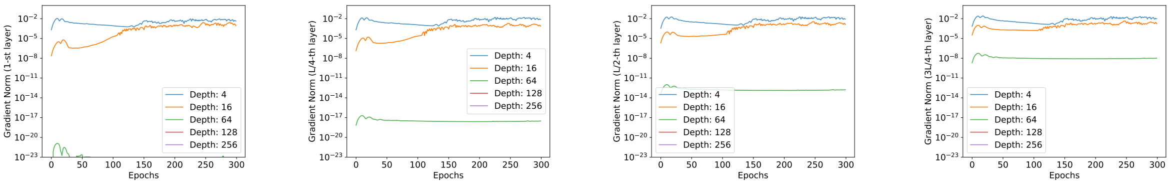
Figure 14. The backward metrics of four layers in ReLU-activated JKNet with different depths in 300 epochs for training on IGB-Tiny19 dataset. We replicate this experiment across five random seeds.

In Figures 14, 15, 16, and 17, we present the backward metrics of JKNet, ResGCN, GCNII, and ReZeroGCN with different depths in early training. We see that, across the early 300 epochs, JKNet exhibits stable backward propagation in its first layer, while the gradient norms of the other layers vanish as the depth increases. Across early 300 epochs, the gradient norms of GCNII vanish as the depths increase, while the gradient norms of ResGCN explode. In contrast, the gradient norms of ReZeroGCN quickly improve during early training.



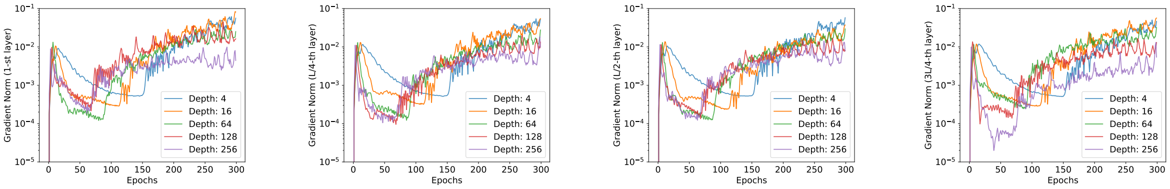
(a) 1-st layer (b) $L/4$-th layer (c) $L/2$-th layer (d) $3L/4$-th layer

Figure 15. The backward metrics of the four layers in ReLU-activated ResGCN with different depths in 300 epochs for training on IGB-Tiny19 dataset. We replicate this experiment across five random seeds. The disappearing lines are caused by surpassing the machine precision.



(a) 1-st layer (b) $L/4$-th layer (c) $L/2$-th layer (d) $3L/4$-th layer

Figure 16. The backward metrics of $W_1^{(l)}$ (see equation 2) in ReLU-activated GCNII with various depths and layers in 300 epochs for training on IGB-Tiny19 dataset. We replicate this experiment across five random seeds. The disappearing lines are caused by surpassing the machine precision.



(a) 1-st layer (b) $L/4$-th layer (c) $L/2$-th layer (d) $3L/4$-th layer

Figure 17. The backward metrics of the four layers of ReLU-activated ReZeroGCN with different depths in 300 epochs for training on IGB-Tiny19 dataset. We replicate this experiment across five random seeds.

H. Limitation and negative social impact

In this paper, we employ signal propagation theory to analyze the curse of depth in Graph Convolutional Networks (GCNs). Additionally, we propose solutions (SPoGInit and ReZeroGCNs) to address signal propagation issues and alleviate the curse of depths in GCNs. Interesting directions for future work include applying signal propagation on the GNNs with attention mechanisms.

This script may provide better guidance for deep graph convolution networks training. It would have potential negative social impact if the models are deployed for illegal usage.