On the Explainability of Graph Convolutional Network With GCN Tangent Kernel

Xianchen Zhou
zhouxianchen13@nudt.edu.cn
Hongxia Wang
wanghongxia@nudt.edu.cn
National University of Defense Technology, Changsha 410073, P.R.C.

Graph convolutional network (GCN) is a powerful deep model in dealing with graph data. However, the explainability of GCN remains a difficult problem since the training behaviors for graph neural networks are hard to describe. In this work, we show that for GCN with wide hidden feature dimension, the output for semisupervised problem can be described by a simple differential equation. In addition, the dynamics of the behavior of output is decided by the graph convolutional neural tangent kernel (GCNTK), which is stable when the width of hidden feature tends to be infinite. And the solution of node classification can be explained directly by the differential equation for a semisupervised problem. The experiments on some toy models speak to the consistency of the GCNTK model and GCN.

1 Introduction

Graph neural networks (GNNs) are widely used in dealing with non-Euclid data (Wu et al., 2021). A typical kind of GNN, the graph convolutional network (GCN; Kipf & Welling, 2016; Wu et al., 2019), achieves great performance in several fields, including society and transportation (Zhou et al., 2020; Wu et al., 2021). Therefore, many researchers focus on the theoretical aspects of GNN, especially GCN, including expressive power (Xu, Hu, Leskovec, & Jegelka, 2018; Loukas, 2020; Chen, Villar, Chen, & Bruna, 2019) and generalization capability (Garg, Jegelka, & Jaakkola, 2020; Scarselli, Tsoi, & Hagenbuchner, 2018; Xu et al., 2020). The explainability of GNN, which often studies the underlying relationship behind predictions, has also gotten the broad attention of numerous experts and scholars (Yuan, Yu, Gui, & Ji, 2020). Since many GNNs are proposed without explaining them, they are treated as a black box and cannot be trusted in critical applications concerning privacy and safety. Therefore, it is necessary to develop

Hongxia Wang is the corresponding author.

Neural Computation **35, 1–26** (2023)
https://doi.org/10.1162/neco_a_01548

explanation techniques to study the causal relationship behind GNN predictions.

Some methods explore GNNs' explainability by identifying important nodes related to their prediction. Gradients and features-based methods (Baldassarre & Azizpour, 2019; Pope, Kolouri, Rostami, Martin, & Hoffmann, 2019) and perturbation-based methods (Ying, Bourgeois, You, Zitnik, & Leskovec, 2019; Luo et al., 2020; Schlichtkrull et al., 2020; Funke et al., 2021) employ different model metrics, including gradients or antiperturbation ability, to indicate the importance of different nodes, while decomposition methods (Schwarzenberg, Hübner, Harbecke, Alt, & Hennig, 2019; Schnake et al., 2020) and surrogate methods (Huang et al., 2020; Vu & Thai, 2020; Zhang et al., 2021) decomposing GNNs or finding a surrogate model to simplify and explain the GNNs. However, most of this work seeks to understand GNN predictions by post hoc explanation, which means the explanations cannot be used to predict GNN output before training.

In this article, we focus on GCN behaviors in dealing with the node classification problem (Kipf & Welling, 2016) and try to interpret GCN, which can be applied to predict its output before training. Since the objective function for training GCN is often nonconvex, it is hard to analyze GCN behavior directly. Recently neural tangent kernel (NTK; Bartlett, Helmbold, & Long, 2018; Arora et al., 2019; Jacot, Gabriel, & Hongler, 2018) has been proposed to analyze deep neural networks including GNNs in different perspectives. Du, Hou, et al. (2019) use the aggregation and combination formulation of infinitely wide GNNs to define the graph NTK (GNTK) at graph level, which can predict the results of graph-level classification problems. Following the definition of GNTK, Huang et al. (2021) defines an NTK of GCN in node level and focus on the trainability of ultrawide GCNs. However, neither of the two works can be applied for analyzing node classification problems directly based on GNTK.

Here, we establish a GCN tangent kernel (GCNTK) in matrix form and use it to analyze the learning dynamics of wide GCN under gradient descent for node classification problems. GCNTK can also predict test nodes' label and help to explain the importance of different training nodes for prediction.

We summarize our contributions as follows:

- *The convergence of GCNTK*. Since the input and output of GCN are in matrix form, the formula of a gaussian process for GCN (GCNGP) is complex. We first give the explicit formula for GCNGP and GCNTK and demonstrate that GCNTK can converge to a fixed form as the GCN's layer width tends to be infinite.
- *The convergence of training loss and stability of GCNTK*. We prove that the loss constrained on the training data set tends to zero, as the width of parameter matrix tends to be infinite. And the GCNTK remains fixed during the training procedure.

- *Predictions for the test nodes' label based on linear dynamics*. We formally obtain the predictions of test nodes' label with infinite-width GCN. We first find that the solution of semisupervised problems mainly depends on the ratio of kernel restricting on the training and test data set. The prediction on the test nodes and the impact of training nodes can be interpreted well by the GCNTK.

2 Related Work

2.1 GNNs and GCNs. GNNs learn task-specific node/edge/graph representations via hierarchical iterative operators and obtain great success in graph learning tasks. A classical GNN consists of aggregation and combination operators, which gather information from neighbors iteratively. GCN, as a typical GNN, defines the convolution on the spectral domain and applies a filter operator on the feature components. Its aggregation function can be treated as a weighted summation of neighbors' feature. Among this, a specific GCN proposed in Kipf and Welling (2016) uses a 1-localized ChebNet to define convolution and obtain the model in equation 3.1 with the bias term, which obtains significant advantages in dealing with node classification problems.

2.2 Explainability of GNNs. The explainability of GNNs can be explored by identifying important nodes related to GNN's prediction. For example, SA and guided BP (Baldassarre & Azizpour, 2019) use gradient square values as the importance and contributions of different nodes, while Grad-CAM (Pope et al., 2019) maps the final layer to the input nodes space to generate the importance. GNN perturbation methods including GNNExplainer (Ying et al., 2019), PGExplainer (Luo et al., 2020), GraphMask (Schlichtkrull, De Cao, & Titov, 2020), and ZORRO (Funke, Khosla, & Anand, 2021) consider the influence of node perturbations on predictions. Surrogate and decomposition models, including GraphLime (Huang et al., 2020), PGM-explainer (Vu & Thai, 2020), Relex (Zhang, Defazio, & Ramesh, 2021), LRP (Baldassarre & Azizpour, 2019), and GNN-LRP (Schnake et al., 2020), explain GNN by employing a simple and interpretable surrogate model to approximate the predictions. However, most of these models explain the GNN afterward and GNN is still a black box to some extent. It means these models can provide an explanation of GNN's output but cannot predict the results before training.

Starting from a spectral GCN, a classical and special GNN, this article addresses GCN's interpretability and analyzes its causal relationship between the predictions and training nodes, which help to predict GCN's output beforehand.

2.3 Neural Tangent Kernel. Based on the gaussian process (GP; Neal, 1995; Lee et al., 2018; de G. Matthews, Rowland, Hron, Turner, &

Ghahramani, 2018) property of deep neural networks, Jacot et al. (2018) introduce the neural tangent kernel and describe the exact dynamics of a fully connected network's output through gradient flow training in an overparameterized situation. This initial work has been followed by a series of studies, including the exacter description (Arora et al., 2019; Lee et al., 2019), generalization for different initialization (Liu, Zhu, & Belkin, 2020; Sohl-Dickstein, Novak, Schoenholz, & Lee, 2020), and NTK for different structures of neural networks (Arora et al., 2019; Li et al., 2021; Luo, Xu, Ma, & Zhang, 2021).

As for graph deep learning, Du, Hou, et al. (2019) considered graph-supervised problems and defined graph NTK (GNTK) in graph level based on GNN's aggregation function, which can be used to predict an unlabeled graph. Immediately after Du, Hou, et al. (2019), Huang et al. (2021) defined GNTK for GCN and conducted research on how to deepen GCN. Since the GNTK formulation in Huang et al. (2021) is based on training nodes, the predictions of testing nodes cannot be obtained directly. In this article, we derive a matrix form of GCN tangent kernel (GCNTK) and provide an explicit form to predict the behavior of testing nodes directly.

3 Preliminaries

Let $G = (V, E, X)$ be a graph with N vertices, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ represents N vertices or nodes. The adjacency matrix $A = (a_{ij})_{N\times N}$ represents the link relation. $X \in \mathbb{R}^{N\times d_0}$ is the node feature matrix. A typical semisupervised problem defined on graph is node classification. Assume that there exist k classes of nodes, and each node v_i in graph G has an associated one-hot representation label $y_i \in \mathbb{R}^k$. However, only some of the nodes in $\mathcal{V}$ denoted by $\mathcal{V}_{train}$ are annotated. The task of node classification is to predict the labels for the unlabeled nodes in $\mathcal{V}_{test}$.

First, we give some notations in matrix form. Assume that the index of a training and a test set is $\mathcal{I}_{train} = \{i|v_i \in \mathcal{V}_{train}\}$ and $\mathcal{I}_{test} = \{i|v_i \in \mathcal{V}_{test}\}$, respectively. For an arbitrary matrix $X = (X_{ij})_{N\times d}$, denote the ith row of matrix X by X_i and the jth column by $X_{:,j}$. Let $I_{train} \in \mathbb{R}^{N\times N}$ be a diagonal matrix where the training index is 1 and the other index set is 0, with a similar definition for I_{test}. $X_{train} = I_{train}X \in \mathbb{R}^{N\times d}$ represents the X constraining on the training set by rows, with a similar definition for X_{test}. For an arbitrary matrix $A \in \mathbb{R}^{N\times N}$, $A_{train,test} = I_{train}AI_{test}$ represents A constraining on the training index by rows and test index by columns.

A GCN $f : \mathbb{R}^{N\times d_0} \longrightarrow \mathbb{R}^{N\times d_{L+1}}$ is a nonlinear transform defined by $H^{L+1} = f(X^0)$, which can be expressed in the recurrent form,

$$\begin{cases} H^{l+1} = AX^l W^{l+1} + B^{l+1} \\ X^{l+1} = \phi(H^{l+1}) \end{cases}, \quad l = 0, \ldots, L, \tag{3.1}$$

with parameters

$$\begin{cases} W^l_{i,j} = \frac{\sigma_\omega}{\sqrt{d_{l-1}}}\omega^l_{i,j} \\ B^l_{i,j} = \sigma_b \beta^l_{i,j} \end{cases}, \tag{3.2}$$

where ϕ represents the activation function. $X^l \in \mathbb{R}^{N\times d_l}$ is the l layer output, and $X^0 = X \in \mathbb{R}^{N\times d_0}$ is the initial node feature matrix of G. $W^{l+1} \in \mathbb{R}^{d_l \times d_{l+1}}$, $B^{l+1} \in \mathbb{R}^{N\times d_{l+1}}$. $\omega^l_{ij}, \beta^l_{i,j}$ are trainable variables drawn independent and identically distributed (i.i.d.) from a standard gaussian with $\omega^l_{ij}, \beta^l_{i,j} \sim \mathbb{N}(0, 1)$ at initialization. The weight and bias variance σ_ω and σ_b are predefined constant variances. The parameterization, equation 3.2, is nonstandard, and we refer to it as "NTK parameterization" (Jacot et al., 2018; Arora et al., 2019; Liu et al., 2020; Littwin, Galanti, Wolf, & Yang, 2020). Unlike the standard parameterization (Sohl-Dickstein, Novak, Schoenholz, & Lee, 2020; Franceschi et al., 2021) that leads to a divergent NTK, the NTK parameterization has a width-dependent scaling factor $\frac{1}{\sqrt{d_{l-1}}}$ in each layer and thus can normalize the backward dynamics by a convergent NTK.

Define $\theta^l \equiv \mathrm{vec}(\{\omega^l_{ij}, \beta^l_{ij}\})$, which is a column-wise stacked $(N + d_{l-1})\ d_l \times 1$ vector of all parameters in layer l. Let $\theta = \mathrm{vec}(\cup_{l=1}^{L+1}\theta^l)$ denote all the network parameters, $\theta^{\leq l_0} = \mathrm{vec}(\cup_{l=1}^{l_0}\theta^l)$, with similar definitions for $\theta^{\geq l_0}$. Let θ_t denote the network parameters at time t, and θ_0 represents the initial values. Combined with the current parameter θ_t, the output of GCN is denoted by the function $f(X, \theta_t) = H^{L+1}(X, \theta_t) \in \mathbb{R}^{N\times d_{L+1}}$, where $d_{L+1} = k$ represents the number of classes of nodes. To simplify the representation, we use $f(X) = f(X, \theta)$, $f_t(X) = f(X, \theta_t)$.

We use the loss $\mathcal{L}$ on the labeled nodes for learning the θ by the gradient descent (GD),

$$\mathcal{L} = 1/2 \sum_{i\in\mathcal{I}_{train}} \ell\left(f(X)_{i,:}, y_i\right), \tag{3.3}$$

where $f(X)_{i,:}$ represents the i row of GCN output.

Then the square loss function equation 3.3, can be written as

$$\mathcal{L} = 1/2\|f(X)_{train} - Y_{train}\|_2^2 = 1/2\|I_{train}(f(X) - Y)\|_2^2. \tag{3.4}$$

Here $Y \in \mathbb{R}^{N\times k}$ is the matrix of N labels in one-hot representation. Although the label of the test nodes, Y_{text} cannot obtained, the loss on the training set slice can be represented directly.

Next, since the gradient flow of GCNs involves the gradient of the matrix, we give its definition as follows. The vectorization of a matrix X is

$$\mathrm{vec}(X) = [X_{11}, \ldots, X_{m1}, X_{12}, \ldots, X_{m2}, \ldots, X_{1n}, \ldots, X_{mn}]^\top \text{ (mn} \times 1). \tag{3.5}$$

We define the derivative of matrix $F \in \mathbb{R}^{p\times q}$ with respect to matrix $X \in \mathbb{R}^{m\times n}$ by vector representation as

$$\nabla_X F = \frac{\partial F}{\partial X} = \frac{\partial\,\mathrm{vec}(F)}{\partial\,\mathrm{vec}(X)} \in \mathbb{R}^{pq\times mn}. \tag{3.6}$$

Let η be the learning rate of GD. Applying the gradient flow on the GCN, the optimization procedure can be written as

$$\dot{\theta}_t = -\eta\nabla_\theta\mathcal{L} = -\eta\nabla_\theta f_t(X)^T\nabla_{f_t(X)}\mathcal{L}. \tag{3.7}$$

Let $\dot{f}_t(X) = \dot{f}_t(\mathrm{vec}(X)) \in \mathbb{R}^{Nd_{L+1}\times 1}$ be a vector representation; then

$$\dot{f}_t(X) = \nabla_\theta f_t(X)\dot{\theta}_t = -\eta\nabla_\theta f_t(X)\nabla_\theta f_t(X)^\top\nabla_{f_t(X)}\mathcal{L} \in \mathbb{R}^{Nd_{L+1}\times 1}. \tag{3.8}$$

Definition 1. *Similar to the definition of NTK in Jacot et al. (2018), the GCNTK Θ_t is defined by*

$$\begin{aligned}\Theta_t = \Theta_t(X,X) &= \nabla_\theta f_t(X)\nabla_\theta f_t(X)^\top \\ &= \Sigma_{l=1}^{L+1}\nabla_{\theta^l} f_t(X)\nabla_{\theta^l} f_t(X)^\top \in \mathbb{R}^{Nd_{L+1}\times Nd_{L+1}}.\end{aligned} \tag{3.9}$$

Then equation 3.8 can be written as

$$\dot{f}_t(X) = -\eta\Theta_t(X,X)\nabla_{f_t(X)}\mathcal{L}. \tag{3.10}$$

Here $\nabla_{f_t(X)}\mathcal{L} \in \mathbb{R}^{Nd_{L+1}\times 1}$ is the gradient of the loss with respect to the output matrix, and $\nabla_\theta f_t(X)^\top \in \mathbb{R}^{|\theta|\times Nd_{L+1}}$ is the gradient of the output with respect to θ at time t.

Equations 3.7 and 3.8 are two differential equations that describe the evolution of parameters and output, respectively. The behavior of output depends on the time-dependent GCNTK Θ_t. However, Θ_t depends on the random draw of θ_t, which makes the differential equations hard to solve. This article assumes that the width tends to infinity to obtain the convergence of Θ_t.

4 Main Results

4.1 Convergence of GCNTK with Respect to Width d. GNTK (Du, Hou, et al., 2019; Huang et al., 2021) gives an NTK formula on GNNs based on the distinct node and the dynamics behavior on the training set. However, for the semisupervised problem on the graph, the convergence of GCNTK in matrix form has not been researched. Since the formula of

GCNTK involves the complex gradient computation of matrix, it has not been given exactly. Therefore, we first focus on the convergence of GCNTK.

Theorem 1. *For a GCN defined by equation 3.1 under the NTK parameterization, the GCNTK at Θ_0 defined by equation 3.9 converges in probability to a deterministic limiting kernel*

$$\Theta_0 \longrightarrow \Theta, \tag{4.1}$$

as the layers width $d_1, d_2, \ldots, d_{L+1} \longrightarrow \infty$.

The proof details of multi-output GPs and NTK are in appendix A. We give the proof sketch as follows.

In order to prove theorem 1, denote $A \otimes B$ as a Kronecker product of A and B. We first show that the output of GCNs at each layer is also in correspondence with a certain class of multioutput GPs defined by

$$H^l \sim GP\left(\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}, \mathcal{K}^l(X, X') \otimes I_{d_l}\right), l = 1, \ldots, L+1, \tag{4.2}$$

where

$$\begin{aligned} \mathcal{K}^l(X, X') &= \sigma_b^2 * I + \sigma_w^2 \mathbb{E}(AX^{l-1})(AX'^{(l-1)})^\top \\ &= \sigma_b^2 * I + \sigma_w^2 \mathbb{E}_{H^{l-1} \sim GP(\vec{0}, \mathcal{K}^{l-1}(X,X') \otimes I_{d_l})} \\ &\quad \times (A\phi(H^{l-1}(X)))(A\phi(H^{l-1}(X')))^\top, \end{aligned} \tag{4.3}$$

with base case

$$\mathcal{K}^1(X, X') = \sigma_b^2 * I + \sigma_w^2 \frac{1}{d_0} AX(AX')^\top, l = 1, 2, \ldots, L+1. \tag{4.4}$$

According to equation 4.2, we have the convergence of GCNTK:

$$\Theta_0 \longrightarrow \Theta^L(X, X)^\top \overset{\Delta}{=} \Theta, \quad \text{as } d_l \longrightarrow \infty, l = 1 \cdots, L+1. \tag{4.5}$$

Here

$$\begin{aligned} \Theta^l(X, X')^\top = I_{d_l} \otimes K^l(X, X') + I_{d_l} \otimes \operatorname{diag}(\dot{\Sigma}^{l-1}(X, X') A \tilde{\Theta}^{l-1}(X, X')^\top A^\top, \\ l = 2, \ldots, L+1, \end{aligned} \tag{4.6}$$

where

$$\dot{\Sigma}^{l-1}(X, X') = \mathbb{E}(\dot{\phi}(H^{l-1}_{:,1}(X)) \circ \dot{\phi}(H^{l-1}_{:,1}(X'))), \tag{4.7}$$

with

$$\Theta^1(X, X')^\top = I_{d_l} \otimes K^1(X, X'). \tag{4.8}$$

$\circ$ is the Hadamard product and $\dot{\phi}$ is the derivative of ϕ.

Remark 1. Jacot et al. (2018) and Yang (2019) show that the NTK of fully connected networks at initialization Θ_0 converges to a deterministic kernel Θ based on the GP property. Theorem 1 shows that the GCNTK also converges at initialization. Compared with the results in Du, Hou, et al. (2019), which define the GNTK from a single node perspective, this article defines the GCNTK in matrix form first and is suitable for further analysis of node classification problems.

4.2 The Behavior of GCNTK with Respect to Time *t*. Furthermore, we focus on the behavior of GCN and GCNTK during the training procedure. We use GCNTK to provide a simple proof of the convergence of GCN under gradient descent on the training nodes. The stability of GCNTK and parameters during the training procedure is guaranteed in our proof. Since Lee et al. (2019) show that both the convergence of NTK and standard parameterization can be obtained by similar proof procedure, we prove the convergence under standard parameterization. Compared with NTK parameterization, standard parameterization is common in the realization of GCN (Kipf & Welling, 2016; Wu et al., 2019), which is defined as

$$\begin{cases} H^{l+1} = AX^l W^{l+1} + B^{l+1} \\ X^{l+1} = \phi(H^{l+1}) \end{cases}, \quad l = 0, \dots, L \tag{4.9}$$

and

$$\begin{cases} W^l_{i,j} = \omega^l_{ij} \sim \mathcal{N}\left(0, \frac{\sigma^2_\omega}{d_l}\right) \\ b^l_{ij} = \beta^l_{ij} \sim \mathcal{N}\left(0, \sigma^2_b\right) \end{cases}. \tag{4.10}$$

Different from the NTK parameterization, the standard NTK kernel (Sohl-Dickstein et al., 2020; Park, Sohl-Dickstein, Le, & Smith, 2019) is defined as

$$\begin{cases} \Theta_t := \Theta_t(X, X) = \frac{1}{d}\nabla_\theta f(X, \theta_t)\nabla_\theta f(X, \theta_t)^T \\ \Theta := \lim_{d\to\infty} \Theta_0 \quad \text{in probability} \end{cases}. \tag{4.11}$$

Theorem 1 has proven that Θ exists. Next, we prove that Θ_t is invariant with respect to t as the width tends to infinity. Before that, we give some assumptions.

1. For the GCN model defined in equations 4.9 and 4.10, the widths in each layers are identical: $d_1 = d_2 = \cdots = d_L = d$.
2. The analytic GCNTK kernel Θ in equation 4.11 is of full rank and positive.
3. The activation function ϕ is Lipschitz continuous and smooth, satisfying

$$|\phi(0)|, \quad \left\|\phi'\right\|_\infty, \quad \sup_{x \neq \tilde{x}} \left|\phi'(x) - \phi'(\tilde{x})\right| / |x - \tilde{x}| = C_2 < \infty, \tag{4.12}$$

$$\sup_{x \neq \tilde{x}} \left|\phi(x) - \phi(\tilde{x})\right| / |x - \tilde{x}| = C_1 < \infty. \tag{4.13}$$

Assumption 1 is easy to satisfy in the limit condition. Assumption 2 holds since the NTK kernel is a multiplication of the derivative. Common activation functions like Relu, and sigmoid satisfy assumption 3.

Under the setting of node classification problem, we have the parameter update formula,

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta f_t(X)^T \nabla_{f_t(X)} \mathcal{L}, \tag{4.14}$$

and the gradient flow equation is

$$\dot{\theta}_t = -\eta \nabla_\theta f_t(X)^T \nabla_{f_t(X)} \mathcal{L}. \tag{4.15}$$

Using the following shorthand,

$$\begin{aligned} f(\theta_t) &\triangleq \operatorname{vec}(f(X, \theta_t)) \in \mathbb{R}^{Nk \times 1}, \\ g(\theta_t) &\triangleq \operatorname{vec}(f(X, \theta_t) - Y) \in \mathbb{R}^{Nk \times 1}, \\ g_{train}(\theta_t) &\triangleq \operatorname{vec}(I_{train}(f(X, \theta_t) - Y)) = I \otimes I_{train} g(\theta_t) \in \mathbb{R}^{Nk \times 1}, \\ J(\theta_t) &\triangleq \nabla_\theta f(\theta_t) \in \mathbb{R}^{Nk \times |\theta|}, \end{aligned} \tag{4.16}$$

we have

$$\theta_{t+1} = \theta_t - \eta J^\top(\theta_t) g_{train}(\theta_t) \tag{4.17}$$

or

$$\dot{\theta}_t = -\eta J^\top(\theta_t) g_{train}(\theta_t). \tag{4.18}$$

Then the behavior of GCN satisfies

$$\dot{f}(\theta_t) = -\eta J(\theta_t) J^\top(\theta_t) I \otimes I_{train} g\left(\theta_t\right). \tag{4.19}$$

Note that $J(\theta_t)J^\top(\theta_t)$ is connected tightly with Θ. Denote $\Theta_{train} = \Theta \cdot (I \otimes I_{train})$. It is easy to prove that the eigenvalues of Θ_{train} are equal to that of Θ except for some zeros. Define

$$\begin{aligned} \lambda_{\min} &= \min_{\lambda > 0} \lambda(\Theta_{train}) > \lambda_{\min}(\Theta) \\ \lambda_{\max} &= \max \lambda(\Theta_{train}) = \lambda_{\max}(\Theta) \\ \eta_{\text{critical}} &= \frac{2}{\lambda_{\min} + \lambda_{\max}}. \end{aligned} \tag{4.20}$$

We first show that the Jacobian is local Lipschitz, where the Lipschitzness constant K is related to parameters A, L, C_1, C_2, and d.

Lemma 1. *There exist a $K > 0$, and N, for every $d \geq N$; the following holds with high probability over random initialization:*

$$\begin{cases} \frac{1}{\sqrt{d}} \|J(\theta) - J(\tilde{\theta})\|_F \leq K \|\theta - \tilde{\theta}\|_2 \\ \frac{1}{\sqrt{d}} \|J(\theta)\|_F \leq K \end{cases}, \quad \forall \theta, \tilde{\theta} \in B\left(\theta_0, d^{-\frac{1}{2}}\right), \tag{4.21}$$

where

$$B\left(\theta_0, R\right) := \left\{\theta : \|\theta - \theta_0\|_2 < R\right\}. \tag{4.22}$$

The proof of lemma 1 is in section B.3 in online appendix B. It shows that in the neighborbood of initialization θ_0, the Jacobian $J(\theta)$ is Lipschitz continuous when d is large enough.

Then for the gradient descent and gradient flow, we have the following main results:

Theorem 2. *Assume assumptions 1, 2, and 3 hold. For $\delta_0 > 0$ and $\eta_0 < \eta_{\text{critical}}$, there exist $R_0 > 0$, $N \in \mathbb{N}$, and $K > 1$, such that for every $d \geq N$, the following holds with probability at least $1 - \delta_0$ over random initialization when applying gradient descent with learning rate $\eta = \frac{\eta_0}{d}$,*

$$\begin{cases} \left\|g_{train}\left(\theta_t\right)\right\|_2 \leq \left(1 - \frac{\eta_0 \lambda_{\min}}{3}\right)^t R_0 \\ \sum_{j=1}^t \left\|\theta_j - \theta_{j-1}\right\|_2 \leq \frac{3KR_0}{\lambda_{\min}} d^{-\frac{1}{2}} \end{cases} \tag{4.23}$$

and

$$\sup_t \|\Theta_0 - \Theta_t\|_F \leq \frac{6K^3R_0}{\lambda_{\min}} d^{-\frac{1}{2}}. \tag{4.24}$$

Theorem 3. *Assume assumptions 1, 2, and 3 hold. For $\delta_0 > 0$ and $\eta_0 < \eta_{\text{critical}}$, there exist $R_0 > 0$, $N \in \mathbb{N}$, and $K > 1$, such that for every $d \geq N$, the following holds with probability at least $1 - \delta_0$ over random initialization when applying gradient flow with learning rate $\eta = \frac{\eta_0}{d}$,*

$$\begin{cases} \left\| g_{train}\left(\theta_t\right) \right\|_2 \leq e^{-\frac{1}{3}\eta_0 \lambda_{\min} t} R_0 \\ \|\theta_t - \theta_0\|_2 \leq \frac{3KR_0}{\lambda_{\min}} \left(1 - e^{-\frac{1}{3}\eta_0 \lambda_{\min} t}\right) d^{-\frac{1}{2}} \end{cases} \tag{4.25}$$

and

$$\sup_t \|\Theta_0 - \Theta_t\|_F \leq \frac{6K^3R_0}{\lambda_{\min}} d^{-\frac{1}{2}}. \tag{4.26}$$

Remark 2. Lee et al. (2019), Du, Lee, Li, Wang, and Zhai (2019), and Allen-Zhu, Li, and Song (2019) show the convergence of an overparameterized, fully connected network and the stability of NTK, while theorems 2 and 3 extend the convergence and stability of GCNTK for a semisupervised problem. As the GCNTK converges at initialization time when the width tends to be infinite, it also remains fixed during training. Different from the supervised problem, the convergence rate of GCN for a semisupervised problem depends on the smallest eigenvalues of the GCNTK constrained on the training nodes. Therefore for the same graph, training GCN with fewer training nodes is faster in general. The proof is in appendix B.

4.3 GCN Explanation Based on Training Dynamics Equation. Under the infinite width assumption, GCNTK remains fixed, and the training dynamics can be computed by the evolution equation 3.8. We can obtain the explanation of GCN with infinite width based on the solution of the training dynamics equation.

Theorem 4. *For a graph convolutional network defined by equation 3.1 in the limits as the layers width $d_1, d_2, \ldots, d_{L+1} \longrightarrow \infty$, the output of GCN is*

$$\frac{\mathrm{d}\,\mathrm{vec}(f_t(X))}{\mathrm{d}t} = -\eta \Theta_{train} \cdot \mathrm{vec}((f_t(X) - Y))), \tag{4.27}$$

where $\Theta_{train} = \Theta \cdot I \otimes I_{train}$.

The solution of equation 3.4

$$\mathrm{vec}(f_t(X) - Y) = e^{-\eta\Theta_{train}t}\mathrm{vec}(f_0(X) - Y). \tag{4.28}$$

The slice of Θ_{train} *constraining on the rows of the testing set and the columns of the training set is* $\Theta_{test,train}$. *Then we have*

$$\mathrm{vec}(f_t(X) - Y)_{train} = e^{-\eta\Theta_{train,train}t}(\mathrm{vec}(f_0(X) - Y))_{train}. \tag{4.29}$$

Similarly, from

$$\begin{aligned}\frac{\mathrm{dvec}(f_t(X))_{test}}{\mathrm{d}t} &= -\eta\Theta_{test,train}\mathrm{vec}(f_t(X) - Y)_{train} \\ &= -\eta\Theta_{test,train}e^{-\eta\Theta_{train,train}t}(\mathrm{vec}(f_0(X) - Y))_{train},\end{aligned} \tag{4.30}$$

we have

$$\begin{aligned}\mathrm{vec}(f_t(X))_{test} = &-\Theta_{test,train}\Theta^{-1}_{train,train}(I - e^{-\eta\Theta_{train,train}t})(\mathrm{vec}(f_0(X) - Y))_{train} \\ &+ \mathrm{vec}(f_0(X))_{test}.\end{aligned} \tag{4.31}$$

Theorem 4 shows that the output label of the test nodes is influenced by two factors of GCNTK, $\Theta_{test,train}$ and $\Theta_{train,train}$. As $t \longrightarrow \infty$, the output label is a linear combination of the label of training nodes. Define

$$\mathcal{M} = \Theta_{test,train}\Theta^{-1}_{train,train}; \tag{4.32}$$

then $\mathcal{M}$ describes the contribution of labeled nodes. Therefore, according to equation 4.31, the influence of training nodes on the testing nodes can be obtained easily.

5 Experiments

Little previous work has provided beforehand explanations of GNN. And to our knowledge, there is no beforehand explanation model investigating the node classification problem. In this article, we compare only the output of GCN and GCNTK and verify the consistency of their predictions. We use synthetic data sets to obtain some conclusions which are implicit in the classical GCN.

Synthetic data sets: In order to verify the prediction correctness in theorem 4, we build some subgraphs with a few nodes that have common patterns in various graph data sets. All the nodes in these graphs except one are labeled with red or green. We use the GCN and GCNTK to

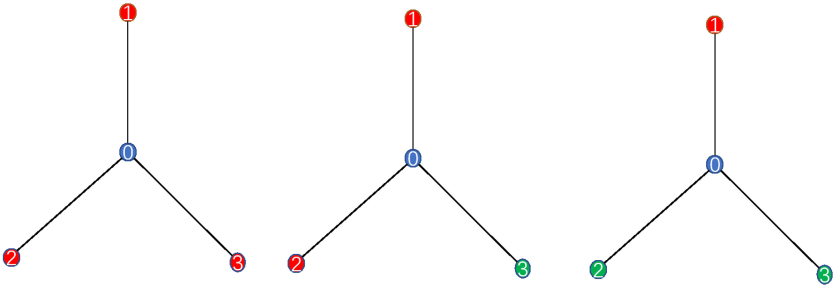

Figure 1: Using GCNTK to predict the color of node 0, the GCNTK ratio $\mathcal{M} = (0.33, 0.33, 0.33)$. Then node 0 in the left, the middle, and the right is predicted using equation 4.31 as red, red, and green, respectively, which is completely consistent with the result of GCN.

predict the label (red or green) of the unlabeled (blue) node, respectively based on labeled nodes, and obtain the contribution of those labeled nodes.

GCN model: We train the node classification model using different types of graphs by the classical GCN in Kipf and Welling (2016) where the feature matrix X is set by the identity matrix. And we only predict the label of one node (in blue).

5.1 Experiments on the Star Graph. Theorem 4 shows that the predicted label is only influenced by the contribution of unlabeled nodes, where the contribution ratio is $\mathcal{M}$ defined by equation 4.32. For instance, if there is no link between the test nodes and training nodes, then $\Theta_{test,train} = 0$. As a result, no node has an effect on the training nodes.

5.2 Common Patterns with Four Nodes. In this section, we compute the contribution ratio $\mathcal{M}$ of all the patterns with four nodes in Figures 1 and 2. As is shown in Table 1, the labeled nodes make different predictions of 0. The ratio displays the importance of the node related to the predicted node. The positive or negative ratio of one training node means that predicted node tends to be the same as or different from that node. And the absolute value of the ratio means the importance of different nodes.

In Figures 2c and 2f, all the labeled nodes make the same positive contribution to the predicted nodes. Therefore, the largest number of those nodes with the same color decide the color of node 0. In addition, in Figures 2g, 2e, and 2h, different nodes make different positive contribution to node 0. And in Figures 2g, 2i, and 2j, some nodes even make negative contribution to node 0. We can find that node 2 in Figure 2g, 1 in Figure 2i, and 2 in Figure 2j make the greatest contribution to the predicted node 0.

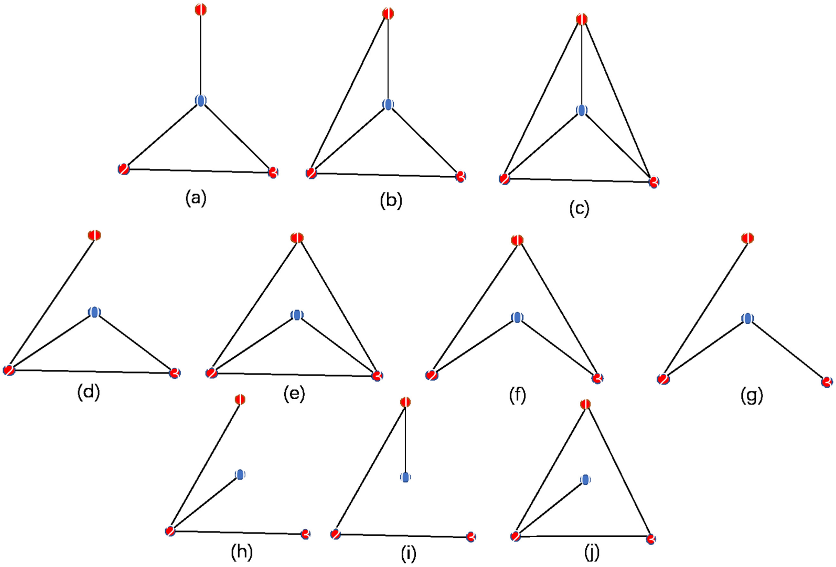

Figure 2: Different patterns with four nodes. Red nodes are training nodes with labels, and the blue node is the unlabeled node to be predicted.

Table 1: The Contribution Ratio $\mathcal{M}$ of Different Graphs.

Number	$\mathcal{M}$	Number	$\mathcal{M}$
a	(0.38, 0.31, 0.31)	b	(0, 1, 0)
c	(0.33, 0.33, 0.33)	d	(0, 0, 1)
e	(0.10, 0.45, 0.45)	f	(0.33, 0.33, 0.33)
g	(−0.34, 0.74, 0.60)	h	(0.29, 0.42, 0.29)
i	(1.5, −0.85, 0.35)	j	(−0.16, 1.32, −0.16)

5.3 A Special Case. In this section, we display an interesting experiment that is inconsistent with the intuition that similar neighbors should have the same labels. The graph in Figure 3 shows that different colored nodes connect with each other, but the node with the same color is separate. Then we use the labeled node 1 to 7 to predict the unlabeled node 0.

6 Conclusion

Graph convolutional networks (GCNs) are widely studied and perform well for node classification problems. However, the causal relationship under the predictions is unknown and thus restricts their applications in the areas of security and privacy. To interpret GCN, we assume that the GCN

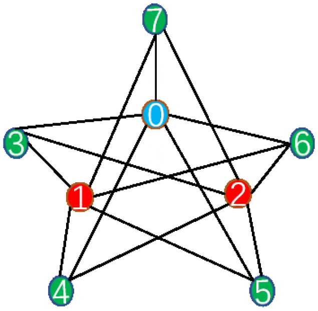

Figure 3: Node 0 is connected with all the green nodes, but GCN still predicts node 0 as red. GCNTK can be used to explain the results. Since the contribution ratio of training nodes is $\mathcal{M} = (0, 0, 0, 0, 0, 0.5, 0.5)$, only red nodes contribute to node 0′s label.

has an infinite width. We define the GCNTK to analyze the GCN training procedure and predict its output. Firstly, we prove that the GCNTK converges and is stable as the width tends to infinite. Then we find for GCN, with an infinite width, that the output value of unlabeled nodes can be predicted by a linear combination of the training nodes′ label. The coefficients of training nodes can be computed by GCNTK and imply the importance of training nodes to the unlabeled nodes. Finally, we conduct experiments on synthetic data sets including common patterns in small model graphs to demonstrate the effectiveness of GCNTK.

Appendix A: Computing GCNGP and GCNTK

Similar to Lee et al. (2018) and Arora et al. (2019), GCNGP can be written as a recursive formula. Let the activation function ϕ be a differentiable function. Let X and X' be two inputs in $\mathbb{R}^{N\times d_1}$. Denote $H^l = [H_1^l, H_2^l, \ldots, H_{d_l}^l]$, where

$$H_j^l = \begin{pmatrix} h_{1j}^l \\ \vdots \\ h_{Nj}^l \end{pmatrix} \tag{A.1}$$

is a column vector. We have

$$\begin{pmatrix} h_{1j}^{l+1} \\ \vdots \\ h_{Nj}^{l+1} \end{pmatrix} = \begin{pmatrix} \Sigma_{m=1}^{d_l} (AX^l)_{1m} W_{mj} + b_{1j} \\ \vdots \\ \Sigma_{m=1}^{d_l} (AX^l)_{Nm} W_{mj} + b_{Nj} \end{pmatrix} = \begin{pmatrix} \langle (AX^l)_{1,:}, W_{:,j} \rangle + B_{1j} \\ \vdots \\ \langle (AX^l)_{N,:}, W_{:,j} \rangle + B_{Nj} \end{pmatrix}. \tag{A.2}$$

Based on the central limit theorem, Lee et al. (2018) show that for each h_{ij}^{l+1}, $i = 1, \dots N$, $j = 1, \dots d_{l+1}$, $h_{ij}^{l+1} \sim GP(0, K_i)$. Note that the gaussian process h_{ij}^{l+1} has the same and independent K_i with respect to j. We need to establish the relation of H_j^{l+1} with respect to column element.

According to the intrinsic coregionalization model for the multioutput gaussian process theory, we have

$$H_j^l \sim GP\left(\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}, \mathcal{K}^l(X, X')\right),$$

$$H \sim GP\left(\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}, \mathcal{K}^l(X, X') \otimes I_{d_l}\right), \tag{A.3}$$

where $\mathcal{K}^l(X, X') = \operatorname{cov}(H_j^l(X), H_j^l(X'))$ is a matrix that represents the covariance of different dimension. For H_j,

$$\begin{aligned} \mathcal{K}^l(X, X')_{mn} &= \mathbb{E}(h_{mj}^l(X) h_{nj}^l(X')) \\ &= \sigma_b^2 \delta(m - n) + \sigma_w^2 \mathbb{E}((AX^{l-1})m, : (AX'^{l-1})n, :^\top). \end{aligned} \tag{A.4}$$

Then we write $\mathcal{K}^l(X, X')$ in matrix form based on equation A.4.

$$\begin{aligned} \mathcal{K}^l(X, X') &= \sigma_b^2 * I + \sigma_w^2 \mathbb{E}(AX^{l-1})(AX'^{(l-1)})^\top \\ &= \sigma_b^2 * I + \sigma_w^2 \mathbb{E}_{H^{l-1} \sim GP(\vec{0}, \mathcal{K}^{l-1}(X, X') \otimes I_{d_l})} \\ &\quad \times (A\phi(H^{l-1}(X)))(A\phi(H^{l-1}(X')))^\top, \end{aligned} \tag{A.5}$$

with base case

$$\mathcal{K}^l(X, X') = \sigma_b^2 * I + \sigma_w^2 \frac{1}{d_0} AX(AX')^\top. \tag{A.6}$$

Let

$$J^l(X) = \nabla_{\theta^{\leq l}} H^l(X) = [\nabla_{\theta^l} H^l(X), \nabla_{\theta^{<l}} H^l(X)]; \tag{A.7}$$

then

$$J^l(X) J^l(X')^\top = \nabla_{\theta^l} H^l(X) \nabla_{\theta^l} H^l(X')^\top + \nabla_{\theta^{<l}} H^l(X) \nabla_{\theta^{<l}} H^l(X')^\top \tag{A.8}$$

and

$$\nabla_{\theta^l} H^l(X)\nabla_{\theta^l} H^l(X')^\top = I_{d_l} \otimes [\sigma_\omega^2 * AX^{l-1}(AX'^{l-1})^\top + \sigma_b^2 * I]. \tag{A.9}$$

When $d_1, d_2, \dots, d_{l-1} \longrightarrow \infty$, this term at initial value converges to $I_{d_l} \otimes K^l(X, X')$. Using $H_0^l(X)$ represents the initial value of $H^l(X)$ and denoting $D^{l-1}(X) = \mathrm{diag}(\dot{\phi}(H_0^{l-1}(X)))$, we have;

$$\frac{\partial H_0^l(X)}{\partial H_0^{l-1}(X)} = D^{l-1}(X)(W_0^\top \otimes A). \tag{A.10}$$

Assume $\Theta^{l-1}(X, X')^\top = I_{d_{l-1}} \otimes \tilde{\Theta}^{l-1}(X, X')^\top$; then

$$\begin{aligned}
&\nabla_{\theta^{<l}} H_0^l(X)\nabla_{\theta^{<l}} H_0^l(X')^\top \\
= \;& \frac{\partial H_0^l(X)}{\partial H_0^{l-1}(X)} \nabla_{\theta^{<l}} H_0^{l-1}(X)\nabla_{\theta^{<l}} H_0^{l-1}(X')^\top \frac{\partial H_0^l(X')}{\partial H_0^{l-1}(X')}^\top \\
\longrightarrow \;& \frac{\partial H_0^l(X)}{\partial H_0^{l-1}(X)} \Theta^{l-1}(X, X')^\top \frac{\partial H_0^l(X')}{\partial H_0^{l-1}(X')}^\top \\
= \;& D^{l-1}(X)(W_0^\top \otimes A)\Theta^{l-1}(X, X')^\top (W_0 \otimes A^\top) D^{l-1}(X)^\top \\
\longrightarrow \;& D^{l-1}(X)(I \otimes A\tilde{\Theta}^{l-1}(X, X')^\top A^\top) D^{l-1}(X)^\top \\
\longrightarrow \;& I \otimes \mathrm{diag}(\dot{\Sigma}^{l-1}(X, X'))(I \otimes A\tilde{\Theta}^{l-1}(X, X')^\top A^\top) \\
= \;& I \otimes \mathrm{diag}(\dot{\Sigma}^{l-1}(X, X')A\tilde{\Theta}^{l-1}(X, X')^\top A^\top.
\end{aligned} \tag{A.11}$$

Denote

$$\dot{\Sigma}^{l-1}(X, X') = \mathbb{E}(\dot{\phi}(H_{0,1}^{l-1}(X)) \circ \dot{\phi}(H_{0,1}^{l-1}(X'))). \tag{A.12}$$

Therefore,

$$\Theta^l(X, X')^\top = I_{d_l} \otimes K^l(X, X') + I_{d_l} \otimes \mathrm{diag}(\dot{\Sigma}^{l-1}(X, X')A\tilde{\Theta}^{l-1}(X, X')^\top A^\top. \tag{A.13}$$

Appendix B: Convergence of GCNTK to Its Linearization, and Stability of GCNTK

In this section, we give a simple proof of the global convergence of GCNTK restricting on the training data set under gradient descent and gradient flow. With a subtle difference with the NTK initialization, we give the proof procedure based on standard parameterization. The GCN is generated with standard parameterization by equations 4.9 and 4.10.

B.1 Proof of Theorem 3. Since the parameter at initialization is randomly generated, there exist R_0 and n_0 such that for every $n > n_0$, with probability at $(1 - \delta_0/10)$ over random initialization,

$$\|g_{train}(\theta_0)\| < R_0. \tag{B.1}$$

Let $C = 3\frac{KR_0}{\lambda_{\min}}$ in lemma 1. There exists a large $n_1 > n_0$ such that for every $d > n_1$, equations 4.21 and B.1 hold with probability at least $(1 - \delta_0/5)$ over random initialization. The case that $t = 0$ holds obviously, and we assume equation 4.21 holds for $t = t$. By induction, we can obtain the second formula in equation 4.23,

$$\|\theta_{t+1} - \theta_t\|_2 \leq \eta \left\|J(\theta_t)\right\|_2 \left\|g_{train}(\theta_t)\right\|_2 \leq \frac{K\eta_0}{\sqrt{d}}\left(1 - \frac{\eta_0\lambda_{\min}}{3}\right)^t R_0. \tag{B.2}$$

$$\begin{aligned}\|\theta_{t+1} - \theta_t\|_2 + \cdots + \|\theta_1 - \theta_0\|_2 &\leq \sum_{j=1}^{t+1} \frac{K\eta_0 R_0}{\sqrt{d}}\left(1 - \frac{\eta_0\lambda_{\min}}{3}\right)^{j-1} \\ &= \frac{K\eta_0 R_0}{\sqrt{d}} \frac{1 - \left(1 - \frac{\eta_0\lambda_{\min}}{3}\right)^{t+1}}{1 - \left(1 - \frac{\eta_0\lambda_{\min}}{3}\right)} \\ &= \frac{K\eta_0 R_0}{\sqrt{d}} 3 \frac{1 - \left(1 - \frac{\eta_0\lambda_{\min}}{3}\right)^{t+1}}{\eta_0\lambda_{\min}} \\ &\leq \frac{K\eta_0 R_0}{\sqrt{d}} \frac{3}{\eta_0\lambda_{\min}} \\ &= \frac{3KR_0}{\lambda_{\min}} d^{-\frac{1}{2}}. \end{aligned} \tag{B.3}$$

Therefore, the second formula of equation 4.23 at $t = t + 1$ can be satisfied. Since

$$\|\theta_{t+1} - \theta_0\|_2 \leq \|\theta_{t+1} - \theta_t\|_2 + \cdots + \|\theta_1 - \theta_0\|_2 \leq \frac{3KR_0}{\lambda_{\min}} d^{-\frac{1}{2}}, \tag{B.4}$$

$\theta_{t+1} \in B\left(\theta_0, Cd^{-\frac{1}{2}}\right).$

$$\begin{aligned} &\left\|g_{train}(\theta_{t+1})\right\|_2 \\ &= \left\|I \otimes I_{train} g(\theta_{t+1}) - I \otimes I_{train} g(\theta_t) + I \otimes I_{train} g(\theta_t)\right\|_2 \end{aligned}$$

$$\begin{aligned}
&= \left\| I \otimes I_{train} J(\tilde{\theta}_t)\left(\theta_{t+1} - \theta_t\right) + I \otimes I_{train} g\left(\theta_t\right)\right\|_2 \\
&= \left\| -\eta I \otimes I_{train} J(\tilde{\theta}_t) J\left(\theta_t\right)^T I \otimes I_{train} g\left(\theta_t\right) + I \otimes I_{train} g\left(\theta_t\right)\right\|_2 \\
&= \left\| \left(I \otimes I_{train} I \otimes I_{train} - \eta I \otimes I_{train} J(\tilde{\theta}_t) J\left(\theta_t\right)^T I \otimes I_{train}\right) g\left(\theta_t\right)\right\|_2 \\
&\leq \left\| I \otimes I_{train} - \eta I \otimes I_{train} J(\tilde{\theta}_t) J\left(\theta_t\right)^T \right\|_2 \left\| g_{train}\left(\theta_t\right)\right\|_2 \\
&\leq \left\| I \otimes I_{train} (I - \eta J(\tilde{\theta}_t) J\left(\theta_t\right)^T) \right\|_2 \left(1 - \frac{\eta_0 \lambda_{\min}}{3}\right)^t R_0, \qquad \text{(B.5)}
\end{aligned}$$

where $\tilde{\theta}_t \in B\left(\theta_0, Cd^{-\frac{1}{2}}\right)$ is a linear interpolation between θ_t and θ_{t+1}.

$$\begin{aligned}
&\left\| I - \eta J(\tilde{\theta}_t) J\left(\theta_t\right)^T)\right\|_2 \\
&= \left\| I - \eta J\left(\theta_0\right) J\left(\theta_0\right)^T + \eta J\left(\theta_0\right) J\left(\theta_0\right)^T - \eta J(\tilde{\theta}_t) J\left(\theta_t\right)^T \right\|_2 \\
&= \left\| I - \eta_0 \Theta_0 + \eta J\left(\theta_0\right) J\left(\theta_0\right)^T - \eta J(\tilde{\theta}_t) J\left(\theta_t\right)^T \right\|_2 \\
&= \left\| I - \eta_0 \Theta + \eta_0 \Theta - \eta_0 \Theta_0 + \eta J\left(\theta_0\right) J\left(\theta_0\right)^T - \eta J(\tilde{\theta}_t) J\left(\theta_t\right)^T \right\|_2 \\
&\leq \left\| I - \eta_0 \Theta \right\|_2 + \eta_0 \left\| \Theta - \Theta_0 \right\|_2 + \eta \left\| J\left(\theta_0\right) J\left(\theta_0\right)^T - J(\tilde{\theta}_t) J\left(\theta_t\right)^T \right\|_2 ; \qquad \text{(B.6)}
\end{aligned}$$

then

$$\begin{aligned}
&\left\| I \otimes I_{train} (I - \eta J(\tilde{\theta}_t) J\left(\theta_t\right)^T)\right\|_2 \\
&\leq \left\| I \otimes I_{train} (I - \eta_0) \Theta \right\|_2 + \eta_0 \left\| I \otimes I_{train} (\Theta - \Theta_0) \right\|_2 \\
&\quad + \eta \left\| I \otimes I_{train} (J\left(\theta_0\right) J\left(\theta_0\right)^T - J(\tilde{\theta}_t) J\left(\theta_t\right)^T)\right\|_2 . \qquad \text{(B.7)}
\end{aligned}$$

Since $\eta_0 < \frac{2}{\lambda_{\min} + \lambda_{\max}}$, we have

$$\left\| I \otimes I_{train} (I - \eta_0 \Theta) \right\|_2 = \sigma_{\max} \left(I \otimes I_{train} (I - \eta_0 \Theta) \right) \leq 1 - \eta_0 \lambda_{\min}. \qquad \text{(B.8)}$$

Because $\Theta_0 \to \Theta$, there exists n_3 such that $d > n_3$,

$$\eta_0 \left\| I \otimes I_{train} (\Theta - \Theta_0) \right\|_2 \leq \eta_0 \left\| I \otimes I_{train} (\Theta - \Theta_0) \right\|_F \leq \frac{\eta_0 \lambda_{\min}}{3}. \qquad \text{(B.9)}$$

$$\begin{aligned}
&\left\| I \otimes I_{train} (J\left(\theta_0\right) J\left(\theta_0\right)^T - J(\tilde{\theta}_t) J\left(\theta_t\right)^T)\right\|_2 \\
&= \left\| J\left(\theta_0\right) J\left(\theta_0\right)^T - J\left(\theta_0\right) J\left(\theta_t\right)^T + J\left(\theta_0\right) J\left(\theta_t\right)^T - J(\tilde{\theta}_t) J\left(\theta_t\right)^T \right\|_2
\end{aligned}$$

$$
\begin{aligned}
&= \left\| J(\theta_0)\left[J(\theta_0)^T - J(\theta_t)^T\right] + \left[J(\theta_0) - J(\tilde{\theta}_t)\right] J(\theta_t)^T \right\|_2 \\
&\leq \left\| J(\theta_0)\right\|_2 \left\| J(\theta_0)^T - J(\theta_t)^T \right\|_2 + \left\| J(\theta_0) - J(\tilde{\theta}_t)\right\|_2 \left\| J(\theta_t)^T \right\|_2 \\
&\leq \left\| J(\theta_0)\right\|_F \left\| J(\theta_0)^T - J(\theta_t)^T \right\|_F + \left\| J(\theta_0) - J(\tilde{\theta}_t)\right\|_F \left\| J(\theta_t)^T \right\|_F \\
&\leq K\sqrt{d}K\sqrt{d} \left\| \theta_0 - \theta_t \right\|_2 + K\sqrt{d}K\sqrt{d} \left\| \theta_0 - \tilde{\theta}_t \right\|_2 \\
&= K^2 d \left\| \theta_t - \theta_0 \right\|_2 + K^2 d \left\| \tilde{\theta}_t - \theta_0 \right\|_2 \\
&\leq 2K^2 d \frac{3KR_0}{\lambda_{\min}} d^{-\frac{1}{2}}. \qquad \text{(B.10)}
\end{aligned}
$$

Choose $d \geq (\frac{18K^3R_0}{\lambda_{\min}})^2$:

$$
\begin{aligned}
\left\| I \otimes I_{train}(1 - \eta J(\tilde{\theta})_t J(\theta_t)^T) \right\|_2 &\leq 1 - \eta_0\lambda_{\min} + \frac{\eta_0\lambda_{\min}}{3} + 2\eta_0 K^2 \frac{3KR_0}{\lambda_{\min}} d^{-\frac{1}{2}} \\
&\leq 1 - \frac{\eta_0\lambda_{\min}}{3}. \qquad \text{(B.11)}
\end{aligned}
$$

Therefore,

$$
\left\| g_{train}(\theta_{t+1}) \right\|_2 \leq \left(1 - \frac{\eta_0\lambda_{\min}}{3}\right)^{t+1} R_0. \qquad \text{(B.12)}
$$

We obtain the convergence of Θ_t:

$$
\begin{aligned}
&\left\| \Theta_0 - \Theta_t \right\|_F \\
&= \frac{1}{d} \left\| J(\theta_0) J(\theta_0)^T - J(\theta_t) J(\theta_t)^T \right\|_F \\
&= \frac{1}{d} \left\| J(\theta_0) J(\theta_0)^T - J(\theta_0) J(\theta_t)^T + J(\theta_0) J(\theta_t)^T - J(\theta_t) J(\theta_t)^T \right\|_F \\
&= \frac{1}{d} \left\| J(\theta_0) \left[J(\theta_0)^T - J(\theta_t)^T\right] + [J(\theta_0) - J(\theta_t)] J(\theta_t)^T \right\|_F \\
&\leq \frac{1}{d} \left\| J(\theta_0) \right\|_F \left\| J(\theta_0)^T - J(\theta_t)^T \right\|_F + \frac{1}{d} \left\| J(\theta_0) - J(\theta_t) \right\|_F \left\| J(\theta_t)^T \right\|_F \\
&\leq K^2 \left\| \theta_t - \theta_0 \right\|_2 + K^2 \left\| \theta_t - \theta_0 \right\|_2 \\
&\leq \frac{6K^3R_0}{\lambda_{\min}} d^{-\frac{1}{2}}. \qquad \text{(B.13)}
\end{aligned}
$$

□

B.2 Proof of Theorem 4. There exist R_0 and d_0 such that $d > d_0$ with probability at $(1 - \delta_0/10)$ over random initialization,

$$\|g_{train}(\theta_0)\|_2 \le R_0. \tag{B.14}$$

Let $C = \frac{3KR_0}{\lambda_{\min}}$. Using the same arguments, one can show that there exists n_2 such that $d > n_2$, with probability at least $(1 - \delta_0/10)$,

$$\frac{1}{d} J(\theta)J(\theta)^T \succ \frac{1}{3}\lambda_{\min}\mathbf{Id} \quad \forall \theta \in B\left(\theta_0, Cd^{-\frac{1}{2}}\right). \tag{B.15}$$

Let

$$t_1 = \inf\left\{t : \|\theta_t - \theta_0\|_2 \ge \frac{3KR_0}{\lambda_{\min}} d^{-\frac{1}{2}}\right\}. \tag{B.16}$$

We claim that $t_1 = \infty$. If not, then for all $t < t_1$, $\theta_t \in B(\theta_0, Cd^{-\frac{1}{2}})$ and

$$\Theta_t = \frac{1}{d} J(\theta_t)J(\theta_t)^T \succ \frac{1}{3}\lambda_{\min}\mathbf{Id};$$

thus,

$$\begin{aligned}\frac{d}{dt}\left(\|g_{train}(t)\|_2^2\right) &= -2I \otimes I_{train}\eta_0 g(t)^T \Theta_t I \otimes I_{train} g(t)\\ &\le -\frac{2}{3}\eta_0\lambda_{\min}\|g_{train}(t)\|_2^2. \end{aligned} \tag{B.17}$$

$$\|g_{train}(t)\|_2^2 \le e^{-\frac{2}{3}\eta_0\lambda_{\min}t}\|g_{train}(0)\|_2^2 \le e^{-\frac{2}{3}\eta_0\lambda_{\min}t} R_0^2. \tag{B.18}$$

Note that

$$\begin{aligned}\frac{d}{dt}\|\theta_t - \theta_0\|_2 \le \left\|\frac{d}{dt}\theta_t\right\|_2 &= \frac{\eta_0}{d}\left\|J(\theta_t)\, g_{train}(t)\right\|_2\\ &\le \eta_0 K R_0 e^{-\frac{1}{3}\eta_0\lambda_{\min}t} d^{-1/2}. \end{aligned} \tag{B.19}$$

$$\begin{aligned}\|\theta_t - \theta_0\|_2 &\le \frac{3KR_0}{\lambda_{\min}}\left(1 - e^{-\frac{1}{3}\eta_0\lambda_{\min}t}\right)d^{-\frac{1}{2}} \le \frac{3KR_0}{\lambda_{\min}}\left(1 - e^{-\frac{1}{3}\eta_0\lambda_{\min}t_1}\right)d^{-\frac{1}{2}}\\ &< \frac{3KR_0}{\lambda_{\min}} d^{-\frac{1}{2}}. \end{aligned} \tag{B.20}$$

This contradicts the definition of t_1 and thus $t_1 = \infty$. □

B.3 Proof of Lemma 1.

Theorem 5 (Corollary 5.35 in Vershynin, 2010). *Let $P = P_{N,n}$ be an $N \times n$ random matrix whose entries are independent standard normal random variables. Then for every $t \geq 0$, with probability at least $1 - 2\exp\left(-t^2/2\right)$, one has*

$$\sqrt{N} - \sqrt{n} - t \leq \lambda_{\min}(P) \leq \lambda_{\max}(P) \leq \sqrt{N} + \sqrt{n} + t.$$

Let $\theta = \{W, B\}$ and $\tilde{\theta} = \{\tilde{W}, \tilde{B}\}, \theta_0 = \{W_0, B_0\}$. Therefore, choose a suitable t and a large d; with high probability over random initialization, we have

$$\begin{aligned} &\left\|W^1\right\|, \quad \left\|\tilde{W}^1\right\| \leq \|W_0^1\| + \frac{C}{\sqrt{d}} \leq \sigma_\omega \frac{\sqrt{d} + \sqrt{d_0} + t}{\sqrt{d_0}} + \frac{C}{\sqrt{d}} \leq 3\sigma_\omega \frac{\sqrt{d}}{\sqrt{d_0}}, \\ &\left\|W_0^l\right\|, \left\|\tilde{W}_0^l\right\| \leq 3\sigma_\omega \text{ for } 2 \leq l \leq L+1. \end{aligned} \tag{B.21}$$

For $l \geq 1$, denote

$$\delta^l(X,\theta) = \frac{\partial f^{L+1}(X,\theta)}{\partial H^l(X)} = \prod_{k=l}^{L} \mathrm{diag}(\sigma'(H^l(X)))(W^{l+1\top} \otimes A) \in \mathbb{R}^{Nk \times Nd}. \tag{B.22}$$

Assume that σ_ω and σ_b are the same level. Then according to the definition of GCN and assumption of ϕ, $\|\phi(H^l(X,\theta))\|_2$ can be controlled by the $\mathcal{O}\left(\prod_{i=1}^{l} C_1 \|W^i\|_2 \|A\|_2\right)$, and $\|\delta^l(X,\theta)\|_2$ can be controlled by $\mathcal{O}\left(\prod_{i=l}^{L} C_2 \|W^i\|_2 \|A\|_2\right)$, where C_1 and C_2 are the Lipschitz continuous and smoothness constants. Therefore, there exists a constant K_1, depending on σ_ω^2, σ_b^2, N, and L such that for $l = 1, \ldots, L$,

$$\begin{aligned} &\|\phi(H^l(X,\theta))\|_2 \leq K_1 d^{1/2}, \quad \left\|\delta^l(X,\theta)\right\|_2 \leq K_1 \\ &\|\phi(H^l(X,\theta)) - \phi(H^l(X,\tilde{\theta}))\|_2 \leq K_1 d^{1/2} (\theta - \tilde{\theta})\|_2 \\ &\left\|\delta^l(X,\theta) - \delta^l(X,\tilde{\theta})\right\|_2 \leq K_1 \|(\theta - \tilde{\theta})\|_2. \end{aligned} \tag{B.23}$$

With high probability over random initialization, we have

$$\begin{aligned} \|J(\theta)\|_F^2 &= \sum_l \left\|J\left(W^l\right)\right\|_F^2 + \left\|J\left(B^l\right)\right\|_F^2 \\ &= \sum_l \left\|I \otimes A\phi(H^l(X,\theta))\delta^l(X,\theta)\right\|_F^2 + \left\|I \otimes I\delta^l(X,\theta)\right\|_F^2 \\ &\leq \sum_l (1 + \|A\|_2^2 K_1^2 d) \left\|\delta^l(X,\theta)\right\|_F^2 \end{aligned}$$

$$
\begin{aligned}
&\leq \sum_l (1 + \|A\|_2^2 K_1^2 d) Nk \|\delta^l(X,\theta)\|_2^2 \\
&\leq \sum_l (1 + \|A\|_2^2 K_1^2 d) K_1^2 Nk \|\theta - \tilde{\theta}\|_2^2 \\
&\leq 2(L+1)(\|A\|_2^2 K_1^2 d) K_1^2 Nk. \qquad \text{(B.24)}
\end{aligned}
$$

$$
\begin{aligned}
\|J(\theta) - J(\tilde{\theta})\|_F^2 &= \sum_l \|I \otimes A\phi(H^l(X,\theta))\delta^l(X,\theta) - I \otimes A\phi((H^l(X,\tilde{\theta})))\delta^l(X,\tilde{\theta})\|_F^2 \\
&\quad + \|\delta^l(X,\theta) - \delta^l(X,\tilde{\theta})\|_F^2 \\
&\leq \sum_l (\|I \otimes A\phi(H^l(X,\theta))\delta^l(X,\theta) - I \otimes A\phi(H^l(X,\theta))\delta^l(X,\tilde{\theta})\|_F^2 \\
&\quad + \|I \otimes A\phi(H^l(X,\theta))\delta^l(X,\tilde{\theta}) - I \otimes A\phi((H^l(X,\tilde{\theta})))\delta^l(X,\tilde{\theta})\|_F^2 \\
&\quad + \|\delta^l(X,\theta) - \delta^l(X,\tilde{\theta})\|_F^2) \\
&\leq \sum_l (\|A\| K_1^4 d + \|A\| K_1^4 d + K_1^2) Nk \|\theta - \tilde{\theta}\|_2^2 \\
&\leq 3(L+1)\|A\| K_1^4 d Nk \|\theta - \tilde{\theta}\|_2^2. \qquad \text{(B.25)}
\end{aligned}
$$

Set $K^2 = \max(2(L+1)(\|A\|_2^2 K_1^2) K_1^2 Nk, 3(L+1)\|A\| K_1^4 Nk)$; then

$$
\begin{aligned}
\frac{1}{\sqrt{d}} \|J(\theta)\|_F &\leq K \\
\frac{1}{\sqrt{d}} \|J(\theta) - J(\tilde{\theta})\|_F &\leq K \|\theta - \tilde{\theta}\|_2. \qquad \text{(B.26)}
\end{aligned}
$$

□

Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant 61977065) and the National Key Basic Research Program (grant 2020YFA0713504).

References

Allen-Zhu, Z., Li, Y., & Song, Z. (2019). A convergence theory for deep learning via over-parameterization. In *Proceedings of the International Conference on Machine Learning* (pp. 242–252).

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R., & Wang, R. (2019). On exact computation with an infinitely wide neural net. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems, 32* (pp. 8139–8148). Red Hook, NY: Curran.

Baldassarre, F., & Azizpour, H. (2019). *Explainability techniques for graph convolutional networks*. arXiv:1905.13686.

Bartlett, P., Helmbold, D., & Long, P. (2018). Gradient descent with identity initialization efficiently learns positive definite linear transformations by deep residual networks. In *Proceedings of theInternational Conference on Machine Learning* (pp. 521–530).

Chen, Z., Villar, S., Chen, L., & Bruna, J. (2019). *On the equivalence between graph isomorphism testing and function approximation with GNNs.* arXiv:1905.12560.

de G. Matthews, A. G., Rowland, M., Hron, J., Turner, R. E., & Ghahramani, Z. (2018). Gaussian process behaviour in wide deep neural networks. In *Proceedings of the International Conference on Learning Representations.*

Du, S., Lee, J., Li, H., Wang, L., & Zhai, X. (2019). Gradient descent finds global minima of deep neural networks. In *Proceedings of the International Conference on Machine Learning* (pp. 1675–1685).

Du, S. S., Hou, K., Salakhutdinov, R. R., Poczos, B., Wang, R., & Xu, K. (2019). Graph neural tangent kernel: Fusing graph neural networks with graph kernels. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems*, *32* (pp. 5723–5733). Red Hook, NY: Curran.

Franceschi, J.-Y., de Bézenac, E., Ayed, I., Chen, M., Lamprier, S., & Gallinari, P. (2021). *A neural tangent kernel perspective of GANs.* arXiv:2016.05566v4.

Funke, T., Khosla, M., & Anand, A. (2021). *Hard masking for explaining graph neural networks.* https://openreview.net/forum?id=uDN8pRAdsoC

Garg, V., Jegelka, S., & Jaakkola, T. (2020). Generalization and representational limits of graph neural networks. In *Proceedings of the International Conference on Machine Learning* (pp. 3419–3430).

Huang, Q., Yamada, M., Tian, Y., Singh, D., Yin, D., & Chang, Y. (2020). Graphlime: *Local interpretable model explanations for graph neural networks.* arXiv:2001.06216.

Huang, W., Li, Y., Du, W., Da Xu, R. Y., Yin, J., Chen, L., & Zhang, M. (2021). *Towards deepening graph neural networks: A GNTK-based optimization perspective.* arXiv:2103.03113.

Jacot, A., Gabriel, F., & Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems, 32* (pp. 8580–8589). Red Hook, NY: Curran.

Kipf, T. N., & Welling, M. (2016). *Semi-supervised classification with graph convolutional networks.* arXiv:1609.02907.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., & Sohl-Dickstein, J. (2018). Deep neural networks as gaussian processes. In *Proceedings of the International Conference on Learning Representations.*

Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., & Pennington, J. (2019). Wide neural networks of any depth evolve as linear models under gradient descent. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems*, *32* (pp. 8572–8583). Red Hook, NY: Curran.

Li, M. B., Nica, M., & Roy, D. M. (2021). *The future is log-gaussian: ReNets and their infinite-depth-and-width limit at initialization*. arXiv:2106.04013.

Littwin, E., Galanti, T., Wolf, L., & Yang, G. (2020). On infinite-width hypernetworks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems*, *33*. Red Hook. NY: Curran.

Liu, C., Zhu, L., & Belkin, M. (2020). On the linearity of large non-linear models: When and why the tangent kernel is constant. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems*, *33*. Red Hook. NY: Curran.

Loukas, A. (2020). *How hard is to distinguish graphs with graph neural networks?* arXiv:2005.06649.

Luo, D., Cheng, W., Xu, D., Yu, W., Zong, B., Chen, H., & Zhang, X. (2020). *Parameterized explainer for graph neural network.* arXiv:2011.04573.

Luo, T., Xu, Z.-Q. J., Ma, Z., & Zhang, Y. (2021). Phase diagram for two-layer ReLU neural networks at infinite-width limit. *Journal of Machine Learning Research*, *22*(71), 1–47.

Neal, R. M. (1995). *Bayesian learning for neural networks*. Lecture Notes in Computer Science 118. Berlin: Springer.

Park, D., Sohl-Dickstein, J., Le, Q., & Smith, S. (2019). The effect of network width on stochastic gradient descent and generalization: an empirical study. In *Proceedings of the International Conference on Machine Learning* (pp. 5042–5051).

Pope, P. E., Kolouri, S., Rostami, M., Martin, C. E., & Hoffmann, H. (2019). Explainability methods for graph convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10772–10781). Piscataway, NJ: IEEE.

Scarselli, F., Tsoi, A. C., & Hagenbuchner, M. (2018). The Vapnik–Chervonenkis dimension of graph and recursive neural networks. *Neural Networks*, *108*, 248–259. 10.1016/j.neunet.2018.08.010

Schlichtkrull, M. S., De Cao, N., & Titov, I. (2020). *Interpreting graph neural networks for NLP with differentiable edge masking*. arXiv:2010.00577.

Schnake, T., Eberle, O., Lederer, J., Nakajima, S., Schütt, K. T., Müller, K.-R., & Montavon, G. (2020). *Higher-order explanations of graph neural networks via relevant walks*. arXiv:2006.03589.

Schwarzenberg, R., Hübner, M., Harbecke, D., Alt, C., & Hennig, L. (2019). *Layer-wise relevance visualization in convolutional text graph classifiers*. arXiv:1909.10911.

Sohl-Dickstein, J., Novak, R., Schoenholz, S. S., & Lee, J. (2020). *On the infinite width limit of neural networks with a standard parameterization.* arXiv:2001.07301.

Vershynin, R. (2010). *Introduction to the non-asymptotic analysis of random matrices*. arXiv preprint arXiv:1011.3027.

Vu, M. N., & Thai, M. T. (2020). *PGM-explainer: Probabilistic graphical model explanations for graph neural networks*. arXiv:2010.05788.

Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., & Weinberger, K. (2019). Simplifying graph convolutional networks. In *Proceedings of the International Conference on Machine Learning* (pp. 6861–6871).

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(1), 4–24. 10.1109/Tnnls.2020.2978386

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018). *How powerful are graph neural networks?* arXiv:1810.0026.

Xu, K., Zhang, M., Li, J., Du, S. S., Kawarabayashi, K.-i., & Jegelka, S. (2020). *How neural networks extrapolate: From feedforward to graph neural networks*. arXiv:2009.11848.

Yang, G. (2019). *Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation.* arXiv:1902.04760.

Ying, R., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J. (2019). GNNExplainer: Generating explanations for graph neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems*, *32*, 9240. Red Hook, NY: Curran.

Yuan, H., Yu, H., Gui, S., & Ji, S. (2020). *Explainability in graph neural networks: A taxonomic survey.* arXiv:2012.15445.

Zhang, Y., Defazio, D., & Ramesh, A. (2021). RelEx: A model-agnostic relational model explainer. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society* (pp. 1042–1049). New York: ACM.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., . . . Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, *1*, 57–81. https://www.sciencedirect.com/science/article/pii/S2666651021000012. 10.1016/j.aiopen.2021.01.001

Received March 22, 2022; accepted July 21, 2022.