Appendix for ”DeepGRAND: Deeper Graph Neural Diffusion”

A TECHNICAL PROOFS

Consider the dynamic given by equation 3 and equation 4. We say that a matrix is postitive if and only if all of its entries are positive. Clearly, $\boldsymbol{A}$ is a postive right-stochastic matrix. We recall the well-known Perron-Frobenius theorem

Theorem 1 (Perron-Frobenius)**.** *Let $\boldsymbol{M}$ be a positive matrix. There is a positive real number r, called the Perron–Frobenius eigenvalue, such that r is an eigenvalue of M and any other eigenvalue λ (possibly complex) in absolute value is strictly smaller than r , $|\lambda| < r$. This eigenvalue is simple and its eigenspace is one-dimensional.*

Proof of Proposition 1. Since $\boldsymbol{A}$ is right-stochastic, its Perron-Frobenius eigenvalue is $\alpha_1 = 1$. Its eigenspace has the basis $\boldsymbol{u}_1 = \{1, 1, \dots, 1\}$. Suppose $\{\alpha_1, \alpha_2, \dots, \alpha_k\}$ is the complex spectrum of $\boldsymbol{A}$. That is, they are all complex eigenvalues of $\boldsymbol{A}$. The matrix $\boldsymbol{A} - \boldsymbol{I}$ has eigenvalues $\beta_i = \alpha_i - 1$ for all $i = \overline{1, k}$, so $\beta_1 = 0$ and $\operatorname{Re}\beta_i < 0$ for all $i = \overline{2, k}$. There exists a basis containing $\boldsymbol{u}_1$ of $\mathbb{C}^n$ comprising of generalized eigenvectors of $\boldsymbol{A}$. By rearranging the vectors if needed, the transition matrix $\boldsymbol{P}$ from the standard basis to this basis satisfies

$$\boldsymbol{P}^{-1}(\boldsymbol{A} - \boldsymbol{I})\boldsymbol{P} = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & \boldsymbol{J}_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \boldsymbol{J}_m \end{pmatrix},$$

where each $\boldsymbol{J}_i$ is a Jordan block associated with some eigenvalue β_i. □

Proof of Proposition 2. Let $\boldsymbol{J} = \boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P}$ and $\boldsymbol{Z}(t) = \boldsymbol{P}^{-1}\boldsymbol{X}(t)$ as in Proposition 1, we can rewrite equation 4 as

$$\frac{\partial \boldsymbol{Z}}{\partial t}(t) = \boldsymbol{J}\boldsymbol{Z}(t),$$

The solution to this matrix differential equation is

$$\boldsymbol{Z}(t) = \exp(t\boldsymbol{J})\boldsymbol{Z}(0),$$

where

$$\exp(t\boldsymbol{J}) = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ & \exp(t\boldsymbol{J}_1) & 0 & \ddots & \vdots \\ & & \exp(t\boldsymbol{J}_2) & \ddots & \vdots \\ & & & \ddots & \vdots \\ & & & & 0 \\ & & & & \exp(t\boldsymbol{J}_m) \end{pmatrix}.$$

Let $\overline{\boldsymbol{Z}}$ be the matrix with the same size as $\boldsymbol{Z}(0)$ obtained by setting every entry in all but the first row of $\boldsymbol{Z}(0)$ be equal to 0. We have

$$\boldsymbol{Z}(t) - \overline{\boldsymbol{Z}} = \boldsymbol{Z}(t) - \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} \boldsymbol{Z}(0) = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ & \exp(t\boldsymbol{J}_1) & 0 & \ddots & \vdots \\ & & \exp(t\boldsymbol{J}_2) & \ddots & \vdots \\ & & & \ddots & \vdots \\ & & & & 0 \\ & & & & \exp(t\boldsymbol{J}_m) \end{pmatrix} \boldsymbol{Z}(0).$$

Denote the size of each Jordan block $\boldsymbol{J}_i$ as n_i, we have well-known equality

$$\exp(t\boldsymbol{J}_i) = e^{\beta_i t}\begin{pmatrix} 1 & t & \frac{t^2}{2} & \dots & \frac{t^{n_i-1}}{(n_i-1)!} \\ & 1 & t & \ddots & \vdots \\ & & 1 & \ddots & \vdots \\ & & & \ddots & \vdots \\ & & & & t \\ & & & & 1 \end{pmatrix}.$$

Hence, every non-zero entry of $\boldsymbol{Z}(t) - \overline{\boldsymbol{Z}}$ has the form $e^{\beta_i t}P(t)$ for some $\beta_i < 0$ and polynomial P. Each entry can thus be bounded above in norm by an exponential term of the form $c_1 e^{-c_2 t}$ for some $c_1, c_2 > 0$. A simple calculation shows that $\|\boldsymbol{X}(t) - \boldsymbol{P}\overline{\boldsymbol{Z}}\|_\infty = \|\boldsymbol{P}(\boldsymbol{Z}(t) - \overline{\boldsymbol{Z}})\|_\infty$ would also be bounded by an exponential term. That is, there exists some $C_1, C_2 > 0$ such that

$$\|\boldsymbol{X}(t) - \boldsymbol{P}\overline{\boldsymbol{Z}}\|_\infty \le C_1 e^{-C_2 t}. \tag{12}$$

Recall that the first column of $\boldsymbol{P}$ is $\boldsymbol{u}^\top = (1, 1, \dots, 1)^\top$, so that

$$\boldsymbol{P}\overline{\boldsymbol{Z}} = \begin{pmatrix} \boldsymbol{Z}(0)_{1,1} & \boldsymbol{Z}(0)_{1,2} & \dots & \boldsymbol{Z}(0)_{1,d} \\ \boldsymbol{Z}(0)_{1,1} & \boldsymbol{Z}(0)_{1,2} & \dots & \boldsymbol{Z}(0)_{1,d} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{Z}(0)_{1,1} & \boldsymbol{Z}(0)_{1,2} & \dots & \boldsymbol{Z}(0)_{1,d} \end{pmatrix}.$$

Let $\boldsymbol{v} = (\boldsymbol{Z}(0)_{1,1}, \boldsymbol{Z}(0)_{1,2}, \dots, \boldsymbol{Z}(0)_{1,d})^\top$ and $\boldsymbol{V} = (\boldsymbol{v}, \boldsymbol{v}, \dots, \boldsymbol{v})^\top$. We can rewrite equation 12 as

$$\|\boldsymbol{X}(t) - \boldsymbol{V}\|_\infty \le C_1 e^{-C_2 t}. \tag{13}$$

The claim that $\boldsymbol{v} \in \mathbb{R}^d$ follows from the fact that it is the limit of real-valued vectors with regards to the $\|\cdot\|_\infty$ norm. ☐

Proof of Proposition 3. Let $\boldsymbol{Y}_i(t) = \|\boldsymbol{X}_i(t)\|^4$, we have

$$\begin{aligned} \boldsymbol{Y}_i'(t) &= 4\|\boldsymbol{X}_i(t)\|^2\langle \boldsymbol{X}_i'(t), \boldsymbol{X}_i(t)\rangle \\ &= 4\|\boldsymbol{X}_i(t)\|^2\langle (\boldsymbol{A} - (1+\epsilon)\boldsymbol{I})\boldsymbol{X}_i(t)\|\boldsymbol{X}_i\|^\alpha, \boldsymbol{X}_i(t)\rangle \\ &= 4\|\boldsymbol{X}_i(t)\|^{2+\alpha}\langle (\boldsymbol{A} - (1+\epsilon)\boldsymbol{I})\boldsymbol{X}_i(t), \boldsymbol{X}_i(t)\rangle \\ &= 4\|\boldsymbol{X}_i(t)\|^{2+\alpha}\langle \boldsymbol{A}\boldsymbol{X}_i(t), \boldsymbol{X}_i(t)\rangle - 4(1+\epsilon)\|\boldsymbol{X}_i(t)\|^{4+\alpha} \end{aligned}$$

Since $\boldsymbol{A}$ is a right-stochastic and radial matrix, we have

$$\langle \boldsymbol{A}\boldsymbol{X}_i(t), \boldsymbol{X}_i(t)\rangle \le \|\boldsymbol{A}\|\|\boldsymbol{X}_i(t)\|^2 = \|\boldsymbol{X}_i(t)\|^2,$$

and

$$-\langle \boldsymbol{A}\boldsymbol{X}_i(t), \boldsymbol{X}_i(t)\rangle \le \|\boldsymbol{A}\|\|\boldsymbol{X}_i(t)\|^2 = \|\boldsymbol{X}_i(t)\|^2,$$

so that

$$-\|\boldsymbol{X}_i(t)\|^2 \le \langle \boldsymbol{A}\boldsymbol{X}_i(t), \boldsymbol{X}_i(t)\rangle \le \|\boldsymbol{X}_i(t)\|^2.$$

Hence, we deduce that

$$4\|\boldsymbol{X}_i(t)\|^{4+\alpha}(-2-\epsilon) \le \boldsymbol{Y}_i'(t) \le 4\|\boldsymbol{X}_i(t)\|^{4+\alpha}(-\epsilon). \tag{14}$$

Multiply both sides of equation 14 with $-\frac{\alpha}{4}\|\boldsymbol{X}_i(t)\|^{-4-\alpha} = -\frac{\alpha}{4}\boldsymbol{Y}_i(t)^{-1-\alpha/4}$, and by noting that $-\frac{\alpha}{4}\boldsymbol{Y}_i'\boldsymbol{Y}_i^{-1-\alpha/4} = (\boldsymbol{Y}_i^{-\alpha/4})'$, we have

$$\alpha(2+\epsilon) \ge (\boldsymbol{Y}_i^{-\alpha/4})' \ge \alpha\epsilon.$$

Integrate from 0 to T and rearranging the appropriate terms, we get

$$(2+\epsilon)\alpha T + \boldsymbol{Y}_i(0)^{-\alpha/4} \ge \boldsymbol{Y}_i(T)^{-\alpha/4} \ge \epsilon\alpha T + \boldsymbol{Y}_i(0)^{-\alpha/4}.$$

Finally, by noting that $\boldsymbol{Y}_i^{\frac{-\alpha}{4}} = \|\boldsymbol{X}_i\|^{-\alpha}$, we easily get the bound equation 10

$$\left((2+\epsilon)\alpha T + \|\boldsymbol{X}_i(0)\|^{-\alpha}\right)^{\frac{-1}{\alpha}} \le \|\boldsymbol{X}_i(T)\| \le \left(\epsilon\alpha T + \|\boldsymbol{X}_i(0)\|^{-\alpha}\right)^{\frac{-1}{\alpha}}.$$

☐

B PHYSICAL INTERPRETATION OF GRAND

Informally, diffusion describes the process by which a diffusing material is transported from a region of higher to lower density through random microscopic motions. A natural example is the process of heat diffusion, which occurs when a hot object touches a cold object. Heat will diffuse between them until both objects are of the same temperature.

GRAND approaches deep learning on graphs as a continuous diffusion process and treat GNNs as discretisations of an underlying PDE. In doing so, the dynamic in GRAND is the same as that in (heat) diffusion problems. We use this viewpoint to consider the over-smoothing issue.

Consider each node in a given graph as a point in space containing some amount of thermal energy. Each pair of points is connected through a heat pipe. The thermal conductivity (also known as the diffusivity) of each pipe is specific to each pair of points. A thermal conductivity of 0 represents two points that are not connected in the graph. The exact formulation of graph diffusion up to equation 4 represents a closed system. That is, the total amount of energy in all nodes is invariant of time.

In node classification problems, nodes within the same class are often hypothesized as sharing strong connection with each other. This is known as the homophily assumption, i.e. 'like attracts like'. For example, friends are more likely to share an interest, and papers from the same research area tends to cite each other. Homophily is a key principle of many real-world networks, and its effect on GNNs has gained traction as a research direction (Zhu et al., 2020; Yan et al., 2021; Chen et al., 2022; Luan et al., 2022; Ma et al., 2022).

In the context of GRAND, we can assume nodes within the same class as sharing a highly conductive heat pipe. As such, thermal energy is transferred effectively between them, quickly bridging any gap in temperature. This allows for rapid clustering of nodes into classes of different thermal energy levels, which can then be passed into a fully connected layer to perform classification.

This interpretation gives a surprising intuitive explanation to the occurrence of the over-smoothing issue in GRAND. If the dynamic carries on for too long, the energy level of every node will exponentially converge to the average thermal energy, given that the graph is sufficiently connected. Hence, we argue that GRAND (and more generally, any GNN based purely on diffusion) is inherently prone to suffer from over-smoothing.

C ON NEURAL ODES

Traditional neural networks such as residual neural networks, normalizing flows, recurrent-neural-networks (RNNs) learn complicated mappings via the composition of multiple transformations to the hidden states. For example, the residual network updates the future hidden state using the following equation:

$$\boldsymbol{h}_{t+1} = \boldsymbol{h}_t + f(\boldsymbol{h}_t, t, \theta) \tag{15}$$

Where $t \in \{1, \dots, T\}$ is the layer index of the neural network and $\boldsymbol{h}_t$ is the hidden state at layer t. This iterative update rule can be seen as the Euler discretisation of a continuous transformation.

Neural ODEs (Chen et al., 2018) are a class of continuous-depth (or continuous-time) neural networks where the transformation step t is infinitesimally small. Specifically, the hidden state $\boldsymbol{h}_t$ is parameterized using a continuous dynamics with respect to time t:

$$\frac{d\boldsymbol{h}_t}{dt} = f(\boldsymbol{h}_t, t, \theta) \tag{16}$$

Where $f(\boldsymbol{h}_t, t, \theta)$ is specified by a neural network parameterized by θ. Starting from the initial state $\boldsymbol{h}(0)$, Neural ODEs learn the final representation $\boldsymbol{h}(T)$ by solving 16 using a numerical integrator (often with an adaptive step-size solver or an adaptive solver for short) given an error tolerance. Integrating 16 from 0 to T in a single forward pass requires the adaptive solver to evaluate $f(\boldsymbol{h}_t, t, \theta)$ at multiple time-steps. The computational complexity of the forward pass is determined by the number of function evaluations.

The only problem with updating the hidden state by solving 16 numerically is that the black-box ODE solver is not a differentiable operation. Therefore, the usual back-propagation method for optimizing traditional neural networks does not work for Neural ODEs.

The adjoint sensitivity method (or adjoint method) is a memory-efficient alternative of the traditional back-propagation method for training Neural ODEs. We denote $\boldsymbol{h}(T)$ as the prediction of Neural ODEs and the loss between $\boldsymbol{h}(T)$ and the ground truth is $\mathcal{L}$. Then, we define the adjoint state as $\boldsymbol{a}(t) = \partial\mathcal{L}/\partial\boldsymbol{h}(t)$, we have:

$$\frac{d\mathcal{L}}{d\theta} = \int_0^T \boldsymbol{a}(t)^T \frac{\partial f(\boldsymbol{h}(t), t, \theta)}{\partial\theta} dt \tag{17}$$

Where the adjoint state $\boldsymbol{a}(t)$ satisfies the following dynamics:

$$\frac{d\boldsymbol{a}(t)}{dt} = -\boldsymbol{a}(t)^T \frac{\partial f(\boldsymbol{h}(t), t, \theta)}{\partial\boldsymbol{h}(t)} \tag{18}$$

Since the adjoint state in 18 can be solved numerically using an ODE solver, the gradient of the loss function $\mathcal{L}$ with respect to θ in 17 can be evaluated. The computational complexity of the backward pass is determined by the number of function evaluations used by the ODE solver to solve for the adjoint state.

D EULER DISCRETIZATION OF GRAND ALSO SUFFERS FROM OVER-SMOOTHING

Recall that the forward Euler discretization of the GRAND dynamic was given by Thorpe et al. (2022) as

$$\boldsymbol{X}(k\delta_t) = \boldsymbol{X}((k-1)\delta_t) + \delta_t(\boldsymbol{A} - \boldsymbol{I})\boldsymbol{X}((k-1)\delta_t), \tag{19}$$

where $1 > \delta_t > 0$ is the fixed step size, $k = 1, 2, \dots, K$ denotes the layers from 1 to K and $\boldsymbol{X}_k := \boldsymbol{X}(k\delta_t)$ is the node feature at the k-th layer.

We mirror our analysis as in Section 3 almost completely. Some calculations will be omitted for clarity. First, we give an analogous definition to 1 for discrete GNNs.

Definition 2. *Let $\boldsymbol{X}_k \in \mathbb{R}^{n\times d}$ denote the feature representation at the k-th layer of some discrete GNN dynamic. $(\boldsymbol{X}_k)$ is said to experience over-smoothing if there exists a vector $\boldsymbol{v} \in \mathbb{R}^d$ and constants $C_1, C_2 > 0$ such that for $\boldsymbol{V} = (\boldsymbol{v}, \boldsymbol{v}, \dots, \boldsymbol{v})^\top$*

$$\|\boldsymbol{X}_k - \boldsymbol{V}\|_\infty \le C_1 e^{-C_2 k}. \tag{20}$$

Utilising Proposition 1, we can show that

Proposition 4. *With the dynamic given by equation 5 and equation 19, $(\boldsymbol{X}_k)$ experiences over-smoothing.*

Proof. Let $\boldsymbol{J} = \boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P}$ and $\boldsymbol{Z}_k = \boldsymbol{P}^{-1}\boldsymbol{X}_k$ as in Proposition 1, we can rewrite equation 19 as

$$\boldsymbol{Z}_k = \boldsymbol{Z}_{k-1} + \delta_t(\boldsymbol{J} - \boldsymbol{I})\boldsymbol{Z}_{k-1} = \left((1-\delta_t)\boldsymbol{I} + \delta_t\boldsymbol{J}\right)\boldsymbol{Z}_{k-1} = \left((1-\delta_t)\boldsymbol{I} + \delta_t\boldsymbol{J}\right)^k \boldsymbol{Z}_0, \tag{21}$$

where

$$\left((1-\delta_t)\boldsymbol{I} + \delta_t\boldsymbol{J}\right)^k = \begin{pmatrix} 1 & 0 & \dots & 0 \\ & \left((1-\delta_t)\boldsymbol{I} + \delta_t\boldsymbol{J}_1\right)^k & \ddots & \vdots \\ & & \ddots & \vdots \\ & & & 0 \\ & & & \left((1-\delta_t)\boldsymbol{I} + \delta_m\boldsymbol{J}\right)^k \end{pmatrix}.$$

Let $\overline{\boldsymbol{Z}}$ be the matrix with the same size as $\boldsymbol{Z}_0$ obtained by setting every entry in all but the first row of $\boldsymbol{Z}_0$ be equal to 0. We have

$$\boldsymbol{Z}_k - \overline{\boldsymbol{Z}} = \begin{pmatrix} 0 & 0 & \dots & 0 \\ & \left((1-\delta_t)\boldsymbol{I} + \delta_t\boldsymbol{J}_1\right)^k & \ddots & \vdots \\ & & \ddots & \vdots \\ & & & 0 \\ & & & \left((1-\delta_t)\boldsymbol{I} + \delta_m\boldsymbol{J}\right)^k \end{pmatrix} \boldsymbol{Z}_0.$$

Each block $(1-\delta_t)\boldsymbol{I}+\delta_t\boldsymbol{J}_1$ has spectral radius lesser than 1. Hence, every non-zero entry of $\boldsymbol{Z}_k-\overline{\boldsymbol{Z}}$ can be bounded above in norm by an exponential term of the form $c_1e^{-c_2t}$ for some $c_1, c_2 > 0$. We can deduce that there exists some $C_1, C_2 > 0$ such that

$$\|\boldsymbol{X}_k - \boldsymbol{P}\overline{\boldsymbol{Z}}\|_\infty \leq C_1 e^{-C_2 k}. \tag{22}$$

Recall that the first column of $\boldsymbol{P}$ is $\boldsymbol{u}^\top = (1, 1, \ldots, 1)^\top$. Set $\boldsymbol{v}$ to be the first row of $\boldsymbol{Z}_0$. We can rewrite equation 22 as

$$\|\boldsymbol{X}_k - \boldsymbol{V}\|_\infty \leq C_1 e^{-C_2 k}.$$

The claim that $\boldsymbol{v} \in \mathbb{R}^d$ follows from the fact that it is the limit of real-valued vectors with regards to the $\|\cdot\|_\infty$ norm. □

E DEEPGRAND PERFORMANCE ON DIFFERENT NUMBERS OF LABELLED NODES PER CLASS

In this section, we provide additional experiment results to complement Table 3. Apart from the test accuracies of DeepGRAND ran on optimal T values, we add an additional column showing DeepGRAND's results when trained with identical T values as were used in the GRAND paper (Chamberlain et al. (2021b)). The experiments suggest that with the same T values, DeepGRAND already outperforms GRAND on most of the benchmarks with limited number of labeled nodes. We observe that with optimal T values, DeepGRAND has much less test accuracy variances compared to all other designs.

Table 4: The means and standard deviations of test accuracies of DeepGRAND-l and other variants of GRAND experimented with different number of labelled nodes per class. Best results are written in bold (Note: The results for GRAND++-l were imported from Thorpe et al. (2022)).

Dataset	# labelled	GRAND-l	GRAND-nl	GRAND++-l	DeepGRAND-l (GRAND setting)	DeepGRAND-l (optimal T)
Cora	20	82.86 ± 1.12	82.72±2.45	82.95 ± 1.37	83.87 ± 0.49	**84.20 ± 0.71**
	10	80.67 ± 2.19	80.51 ± 1.11	80.86 ± 2.99	82.47 ± 1.11	**82.88 ± 0.78**
	5	77.09 ± 3.05	77.68 ± 2.85	77.80 ± 4.46	79.00 ± 1.87	**80.85 ± 1.08**
	2	74.23 ± 5.58	69.44 ± 4.27	66.92 ± 10.04	67.57 ± 8.02	**76.49 ± 1.97**
	1	57.93± 8.09	55.86 ± 10.04	54.94 ± 16.09	63.79 ± 7.14	**70.15 ± 3.25**
Citeseer	20	71.74 ± 2.94	73.16 ± 3.09	73.53 ± 3.31	73.64 ± 1.73	**74.67 ± 0.78**
	10	66.26 ± 4.19	67.84 ± 3.96	72.34 ± 2.42	72.87 ± 2.29	**73.45 ± 0.84**
	5	69.00 ± 3.74	66.90 ± 4.62	70.03 ± 3.63	70.84 ± 3.04	**71.97 ± 1.03**
	2	58.35 ± 8.98	56.35 ± 5.54	64.98 ± 8.31	63.26 ± 3.89	**69.74 ± 2.97**
	1	49.65 ± 8.67	47.32 ± 6.66	**58.95 ± 9.59**	53.06 ± 8.49	58.35 ± 2.51
Pubmed	20	78.42 ± 0.46	75.19 ± 1.77	79.16 ± 1.37	79.23 ± 1.23	**79.50 ± 0.64**
	10	74.10 ± 1.88	74.18 ± 1.99	75.13 ± 3.88	76.95 ± 3.24	**78.93 ± 1.48**
	5	71.05 ± 1.87	72.07 ± 2.15	71.99 ± 1.91	73.37 ± 2.26	**77.09 ± 1.12**
	2	71.44 ± 3.85	65.50 ± 9.49	69.31 ± 4.87	70.75 ± 3.86	**72.32 ± 1.82**
	1	62.41 ± 7.59	63.47 ± 5.47	65.94 ± 4.87	64.96 ± 7.08	**70.03 ± 1.84**
Computers	20	84.04 ± 0.98	83.28 ± 1.24	85.73 ± 0.50	87.27 ± 1.45	**87.12 ± 0.45**
	10	82.34 ± 2.18	81.27 ± 3.02	82.99 ± 0.81	83.79 ± 1.45	**85.73 ± 1.11**
	5	78.69 ± 0.79	79.65 ± 2.02	**82.64 ± 0.56**	82.06 ± 2.02	82.42 ± 0.33
	2	66.21 ± 11.26	65.11 ± 8.31	76.47 ± 1.48	75.26 ± 2.20	**76.57 ± 1.44**
	1	49.80 ± 14.97	47.26 ± 11.23	67.65 ± 0.37	65.35 ± 6.65	**69.33 ± 2.71**
Photo	20	93.22 ± 0.44	91.76 ± 1.51	**93.55 ± 0.38**	93.52 ± 0.40	93.46 ± 0.66
	10	90.80 ± 1.37	89.02 ± 2.54	90.65 ± 1.19	90.59 ± 2.27	**92.29 ± 0.42**
	5	88.06 ± 2.59	88.31 ± 1.63	88.33 ± 1.21	87.75 ± 1.59	**90.45 ± 0.99**
	2	82.60 ± 2.96	80.61 ± 4.43	83.71 ± 0.90	84.59 ± 1.80	**85.09 ± 0.32**
	1	75.05 ± 5.44	76.33 ± 4.79	83.12 ± 0.78	73.50 ± 5.06	**83.27 ± 1.88**
CoauthorCS	20	90.99 ± 0.56	90.59 ± 0.97	90.80 ± 0.34	91.53 ± 0.33	**91.66 ± 0.59**
	10	89.00 ± 2.08	**89.95 ± 0.68**	86.94 ± 0.46	89.39 ± 0.89	89.80 ± 0.70
	5	84.19 ± 3.59	87.01 ± 1.97	84.83 ± 0.84	86.05 ± 4.64	**88.24 ± 0.68**
	2	75.19 ± 3.84	76.66 ± 6.85	76.53 ± 1.85	79.25 ± 3.89	**82.08 ± 3.39**
	1	56.58 ± 8.44	66.44 ± 8.17	60.30 ± 1.50	65.15 ± 6.34	**71.14 ± 1.66**

Table 5: T values of DeepGRAND used for different benchmarks.

	Cora	Citeseer	Pubmed	CoauthorCS	Computers	Photo
DeepGRAND (GRAND setting)	18.29	7.87	12.94	3.12	3.24	3.58
DeepGRAND (optimal T)	20.29	9.87	14.94	6.58	6.25	8.58