APPENDIX

A. Residual Connection

Common residual connection for GNNs and their corresponding GNNs are described below.

Res. Res is composed of multiple residual blocks containing few stacked layers. Taking the initial input of the n-th residual block as $\mathbf{X}_n$, and the stacked nonlinear layers within the residual block as $\mathbf{F}(\mathbf{X})$:

$$\mathbf{X}_{n+1} = \mathbf{F}(\mathbf{X}_n) + \mathbf{X}_n$$

where residual mapping and identity mapping refer to $\mathbf{F}(\mathbf{X})$ and $\mathbf{X}$ on the right side of the above equation, respectively. Inspired by Res, Guohao Li & Matthias Müller(2019) proposed a residual connection learning framework for GCN and called this model ResGCN which can be simply described as follows:

$$\mathbf{H}_k = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}_{k-1}\mathbf{W}_{k-1}\right) + \mathbf{H}_{k-1}$$

InitialRes. InitialRes is proposed for the first time in APPNP, unlike Res that carries information from the previous layer, it constructs a connection to the initial representation $\mathbf{X}_0$ at each layer:

$$\mathbf{X}_{n+1} = (1-\alpha)\mathbf{H}(\mathbf{X}_n) + \alpha\mathbf{X}_0$$

where $\mathbf{H}(\mathbf{X})$ denotes the aggregation operation within one layer. InitialRes ensures that each node's representation retains at least α-size of the initial feature information. Correspondingly, APPNP can be formulated as:

$$\mathbf{H}_k = (1-\alpha)\,\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}_{k-1} + \alpha\mathbf{H}$$

Based on APPNP, GCNII introduces identity mapping from Res to make up for the deficiency in APPNP.

Dense. Dense proposes a more efficient way to reuse features between layers. The input is the outputs of all previous layers of the network and at each layer Dense stitches them together:

$$\mathbf{X}_{n+1} = \mathbf{H}([\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_n])$$

where $[\cdot]$ denotes the concatenation of the feature map for the output of layers 0 to n. Inspired by Dense, DenseGCN applies a similar idea to GCN, i.e., let the output of the k-th layer contains transformations from all previous GCN layers to exploit the information from different GCN layers:

$$\mathbf{H}_k = \mathbf{AGG}_{dense}(\mathbf{H}, \mathbf{H}_1, \ldots, \mathbf{H}_{k-1})$$

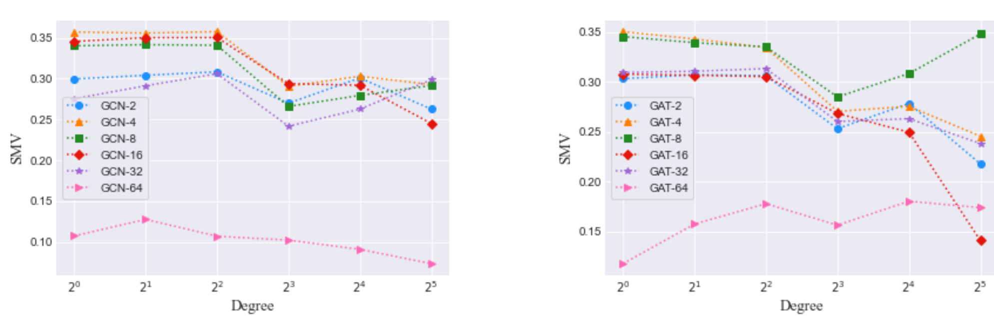
JK. JK is proposed by (Xu et al., 2018). At the last layer, JK sifts from all previous representations $[\mathbf{X}_1, \ldots, \mathbf{X}_N]$ and combines them:

$$\mathbf{X}_{output} = \mathcal{AGG}(\mathbf{X}_1, \ldots, \mathbf{X}_N)$$

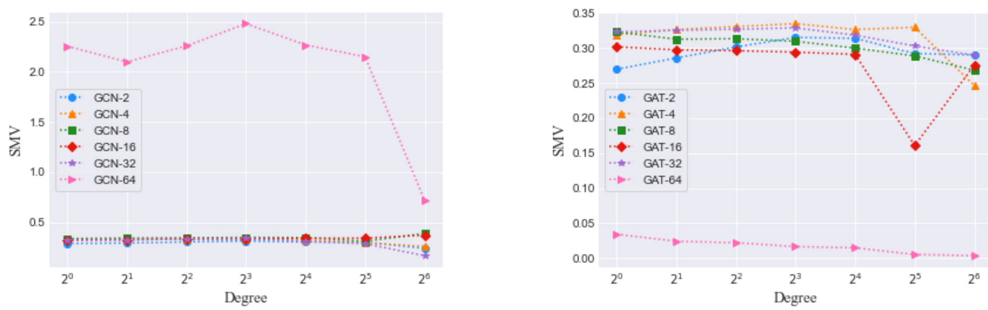
The $\mathcal{AGG}$ operation includes concatenation, Maxpooling and LSTM-attention. When it is introduced to GNN, i.e., JKNet, can be formulated as:

$$\mathbf{H}_{output} = \mathbf{AGG}_{jk}(\mathbf{H}_1, \ldots, \mathbf{H}_{k-1})$$

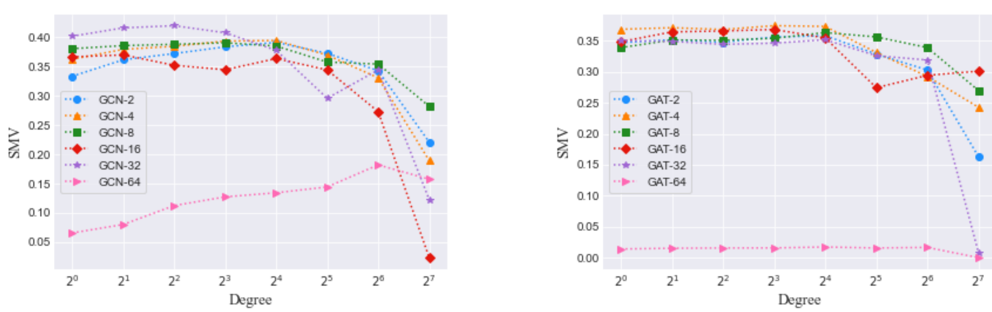
B. SMV for Node Groups of Different Degrees



Citeseer



Cora



Pubmed

C. Derivation of the general formula in the table

ResGCN: We can write the recursive formula for ResGCN in the following form:

$$\mathbf{H}_k = (\mathbf{I} + \mathbf{N})\mathbf{H}_{k-1} \tag{1}$$

In turn, the following form can be obtained by recursion:

$$\mathbf{H}_k = (\mathbf{I} + \mathbf{N})^k \mathbf{H} \tag{2}$$

Using the binomial theorem, we can obtain the general formula for ResGCN as follows:

$$\mathbf{H}_k = \sum_{j=0}^{k} \mathbf{C}_k^j \mathbf{N}^j \mathbf{H} \tag{3}$$

APPNP: According to the recurrence formula of APPNP:

$$\mathbf{H}_k = \alpha \mathbf{H} + (1 - \alpha)\mathbf{N}\mathbf{H}_{k-1}, \tag{4}$$

To obtain the general formula, we can add a term $\mathbf{T}$ to both ends of the equation at the same time:

$$\mathbf{H}_k + \mathbf{T} = (1 - \alpha)\,\mathbf{N}\mathbf{H}_{k-1} + \alpha \mathbf{H} + \mathbf{T} \tag{5}$$

We try to translate the equation into the following form:

$$\mathbf{H}_k + \mathbf{T} = (1 - \alpha)\,\mathbf{N}\,(\mathbf{H}_{k-1} + \mathbf{T})\,. \tag{6}$$

Then we need to make sure that there exits a very $\mathbf{T}$ that satisfies the following equation:

$$(1 - \alpha)\,\mathbf{N}\mathbf{T} = \alpha \mathbf{H} + \mathbf{T}, \tag{7}$$

which can be transformed into the following form:

$$((1 - \alpha)\,\mathbf{N} - \mathbf{I})\,\mathbf{T} = \alpha \mathbf{H}. \tag{8}$$

We can proof that the following lemma:

Lemma 1. *Given that $\alpha \in (0, 1)$, $(1-\alpha)\mathbf{N} - \mathbf{I}$ is invertible.*

Proof. To prove that $(1-\alpha)\mathbf{N} - \mathbf{I}$ is invertible is equivalent to proving it does not have an eigenvalue of 0. Consider the Rayleigh quotient of $(1-\alpha)\mathbf{N} - \mathbf{I}$:

$$\frac{\mathbf{X}^T\left((1-\alpha)\mathbf{N}-\mathbf{I}\right)\mathbf{X}}{\mathbf{X}^T\mathbf{X}} = (1-\alpha)\frac{\mathbf{X}^T\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\right)\mathbf{X}}{\mathbf{X}^T\mathbf{X}} - 1 \tag{9}$$

From spectral graph theory, We can know the following equation holds:

$$\mathbf{X}^T\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{L}\tilde{\mathbf{D}}^{-\frac{1}{2}}\right)\mathbf{X} = \sum_{(v_i,v_j)\in\mathcal{E}}\left(\frac{\mathbf{X}_i}{\sqrt{d_i+1}} - \frac{\mathbf{X}_j}{\sqrt{d_j+1}}\right)^2 > 0. \tag{10}$$

We can decompose $\mathbf{L}$ into $\tilde{\mathbf{D}} - \tilde{\mathbf{A}}$,then we have:

$$\frac{\mathbf{X}^T\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{D}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\right)\mathbf{X}}{\mathbf{X}^T\mathbf{X}} - \frac{\mathbf{X}^T\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\right)\mathbf{X}}{\mathbf{X}^T\mathbf{X}} > 0 \tag{11}$$

which is equivalent to:

$$\frac{\mathbf{X}^T\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\right)\mathbf{X}}{\mathbf{X}^T\mathbf{X}} < \frac{\mathbf{X}^T\mathbf{I}\mathbf{X}}{\mathbf{X}^T\mathbf{X}} = 1 \tag{12}$$

Combining Eq. 9 and Ineq. 12, we can obtain:

$$\frac{\mathbf{X}^T\left((1-\alpha)\mathbf{N}-\mathbf{I}\right)\mathbf{X}}{\mathbf{X}^T\mathbf{X}} < (1-\alpha) - 1 = -\alpha < 0 \tag{13}$$

Therefore, 0 can't be the eigenvalue of $(1-\alpha)\mathbf{N} - \mathbf{I}$. Then $(1-\alpha)\mathbf{N} - \mathbf{I}$ is invertible. □

Since **Lemma 1** holds, We can derive the concrete form of $\mathbf{T}$:

$$\mathbf{T} = \alpha\left((1-\alpha)\mathbf{N} - \mathbf{I}\right)^{-1}\mathbf{H} \tag{14}$$

Thus we can keep recurring from Eq. 6and obtain the following equation:

$$\mathbf{H}_k + \mathbf{T} = \left((1-\alpha)\mathbf{N}\right)^k(\mathbf{H}+\mathbf{T}), \tag{15}$$

which also can be written as:

$$\mathbf{H}_k == \left((1-\alpha)\mathbf{N}\right)^k\mathbf{H} + \left((1-\alpha)\mathbf{N}\right)^k\mathbf{T} - \mathbf{T}. \tag{16}$$

For the second and third term in Eq. 16, We write $(1-\alpha)\mathbf{N}$ as $(1-\alpha)\mathbf{N} - \mathbf{I} + \mathbf{I}$. Then We can use the binomial theorem to write $\left((1-\alpha)\mathbf{N}\right)^k$ as $\sum\limits_{j=0}^{k}\left((1-\alpha)\mathbf{N}-\mathbf{I}\right)^j$ then the Eq. 16 can be written as :

$$\mathbf{H}_k = \left((1-\alpha)\mathbf{N}\right)^k\mathbf{H} + \sum_{j=1}^{k}\left((1-\alpha)\mathbf{N}-\mathbf{I}\right)^j\mathbf{T} \tag{17}$$

Bringing in the specific form of $\mathbf{T}$ and further deriving the general formula of APPNP:

$$\mathbf{H}_k = \left((1-\alpha)\mathbf{N}\right)^k\mathbf{H} + \alpha\sum_{j=0}^{k-1}\left((1-\alpha)\mathbf{N}-\mathbf{I}\right)^j\mathbf{H} \tag{18}$$

$$= (1-\alpha)^k\mathbf{N}^k\mathbf{H} + \alpha\sum_{j=0}^{k-1}\sum_{i=0}^{j}(-1)^{j-i}(1-\alpha)^i\mathbf{N}^i\mathbf{H} \tag{19}$$

D. Proof of theorem 1

For each diagonal element $\lambda_k^{(i)}$ of Λ_k, it is trivial to obtain:

$$0 < \lambda_k^{(i)} < 1.$$

To dervie the general term formula of SNR-GCN, We need proof the following lemma first.

Lemma 2. *Set all the diagonal element of* Λ *satisfy* $0 < \lambda^{(i)} < 1$*, then* $(\Lambda \mathbf{N} + \mathbf{I})$ *is invertible.*

proof. To prove that $\Lambda \mathbf{N} + \mathbf{I}$ is invertible is equivalent to proving that its determinant are not equal to 0. Because all the diagonal element of Λ satisfy $0 < \lambda^{(i)} < 1$,then Λ is invertible, and due to

$$|\Lambda \mathbf{N} + \mathbf{I}| = |\Lambda||\mathbf{N} + \Lambda^{-1}|. \tag{20}$$

Therefore, to prove that its determinant are not equal to 0 is equivalent to proving $|\mathbf{N} + \Lambda^{-1}|$ is not equal to 0, and further equivalent to proving $\mathbf{N} + \Lambda^{-1}$ does not have an eigenvalue of 0.

Consider the Rayleigh quotient of $\mathbf{N} + \Lambda^{-1}$:

$$\mathbf{R}_1 = \frac{\mathbf{X}^T \left(\mathbf{N} + \Lambda^{-1}\right) \mathbf{X}}{\mathbf{X}^T \mathbf{X}} \tag{21}$$

Split Eq. 21, we derive:

$$\mathbf{R}_1 = \frac{\mathbf{X}^T \mathbf{N} \mathbf{X}}{\mathbf{X}^T \mathbf{X}} + \frac{\mathbf{X}^T \Lambda^{-1} \mathbf{X}}{\mathbf{X}^T \mathbf{X}} \tag{22}$$

The second term of Eq. 22 can be easily written as follows:

$$\frac{\mathbf{X}^T \Lambda^{-1} \mathbf{X}}{\mathbf{X}^T \mathbf{X}} = \frac{\sum_{i=1}^{N} {\lambda_i^{(i)}}^{-1} x_i^2}{\sum_{i=1}^{N} x_i^2}.$$

Since $0 < \lambda_i^{(i)} < 1$, therefore ${\lambda_i^{(i)}}^{-1} > 1$, then

$$\frac{\mathbf{X}^T \Lambda^{-1} \mathbf{X}}{\mathbf{X}^T \mathbf{X}} > 1. \tag{23}$$

For the first item, we write its specific form as follows:

$$\frac{\mathbf{X}^T \mathbf{N} \mathbf{X}}{\mathbf{X}^T \mathbf{X}} = \frac{\mathbf{X}^T \left(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}\right) \mathbf{X}}{\mathbf{X}^T \mathbf{X}} \tag{24}$$

From spectral graph theory we know that the following formula holds:

$$\mathbf{X}^T \left(\tilde{\mathbf{D}}^{-\frac{1}{2}} (\mathbf{A} + \mathbf{D}) \tilde{\mathbf{D}}^{-\frac{1}{2}}\right) \mathbf{X} = \sum_{(v_i, v_j) \in \mathcal{E}} \left(\frac{\mathbf{X}_i}{\sqrt{d_i + 1}} + \frac{\mathbf{X}_j}{\sqrt{d_j + 1}}\right)^2 > 0 \tag{25}$$

Further mathematically transforming this formula, we can get the following form:

$$\frac{\mathbf{X}^T \left(\tilde{\mathbf{D}}^{-\frac{1}{2}} (\mathbf{A} + \mathbf{D}) \tilde{\mathbf{D}}^{-\frac{1}{2}}\right) \mathbf{X}}{\mathbf{X}^T \mathbf{X}} = \frac{\mathbf{X}^T \left(\tilde{\mathbf{D}}^{-\frac{1}{2}} \left(\tilde{\mathbf{A}} + \tilde{\mathbf{D}} - 2\mathbf{I}\right) \tilde{\mathbf{D}}^{-\frac{1}{2}}\right) \mathbf{X}}{\mathbf{X}^T \mathbf{X}} \tag{26}$$

$$= \frac{\mathbf{X}^T \left(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}\right) \mathbf{X}}{\mathbf{X}^T \mathbf{X}} + \frac{\mathbf{X}^T \left(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{D}} \tilde{\mathbf{D}}^{-\frac{1}{2}}\right) \mathbf{X}}{\mathbf{X}^T \mathbf{X}} - \frac{2\mathbf{X}^T \tilde{\mathbf{D}}^{-1} \mathbf{X}}{\mathbf{X}^T \mathbf{X}} \tag{27}$$

$$= \frac{\mathbf{X}^T \left(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}\right) \mathbf{X}}{\mathbf{X}^T \mathbf{X}} + 1 - \frac{2\mathbf{X}^T \tilde{\mathbf{D}}^{-1} \mathbf{X}}{\mathbf{X}^T \mathbf{X}} > 0 \tag{28}$$

Further we get the following result:

$$\frac{\mathbf{X}^T \left(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}\right) \mathbf{X}}{\mathbf{X}^T \mathbf{X}} > \frac{2\mathbf{X}^T \tilde{\mathbf{D}}^{-1} \mathbf{X}}{\mathbf{X}^T \mathbf{X}} - 1 \tag{29}$$

It is trivial to obtain:

$$\frac{2\mathbf{X}^T\tilde{\mathbf{D}}^{-1}\mathbf{X}}{\mathbf{X}^T\mathbf{X}} = \frac{2\sum_{i=1}^{N}\left(\mathbf{d}_i+1\right)^{-1}\mathbf{x}_i^2}{\sum_{i=1}^{N}x_i^2} > 0 \tag{30}$$

Combining Eq. 22, Ineq. 23, Ineq. 29 and Ineq. 30, we can get the following inequality:

$$\frac{\mathbf{X}^T\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}+\Lambda^{-1}\right)\mathbf{X}}{\mathbf{X}^T\mathbf{X}} > 0 \tag{31}$$

It can be obtained that the eigenvalue of $\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}+\Lambda^{-1}$ is greater than 0, so 0 is not an eigenvalue of it. Further, $\Lambda\mathbf{N}+\mathbf{I}$ is invertible. □

Now, we proof Theorem 1:

proof. Given the following recursive formula:

$$\mathbf{H}_k = \mathbf{H}_1 + \Lambda_{k-1}\left(\mathbf{H}_1 - \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}\mathbf{H}_{k-1}\right) \tag{32}$$

where $\mathbf{H}_1 = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}\mathbf{H}$, $\mathbf{\Lambda}_k = diag\{\lambda_k^{(1)},...\lambda_k^{(n)}\}$, $\lambda_k^{(i)} \sim Sigmoid(\mathcal{N}(\alpha_k^{(i)},{\beta_k^{(i)}}^2))$. After mathematical transformation, Eq. 32 can be written as

$$\mathbf{H}_k = \left(\mathbf{I}+\Lambda_{k-1}\right)\mathbf{H}_1 - \Lambda_{k-1}\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}\mathbf{H}_{k-1}. \tag{33}$$

Set $\mathbf{N} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$ then Eq. 33 can be abbreviated as

$$\mathbf{H}_k = \left(\mathbf{I}+\Lambda_{k-1}\right)\mathbf{H}_1 - \Lambda_{k-1}\mathbf{N}\mathbf{H}_{k-1}. \tag{34}$$

We tried to modify Eq. 34 to a form that more suitable for obtaining the general term:

$$\mathbf{H}_k + \mathbf{M}_{k-1} = -\Lambda_{k-1}\mathbf{N}\left(\mathbf{H}_{k-1}+\mathbf{M}_{k-1}\right). \tag{35}$$

In order to verify whether there exists such M that satisfies the equation ,we need to solve the following equation:

$$-\Lambda_{k-1}\mathbf{N}\mathbf{M}_{k-1} = \left(\mathbf{I}+\Lambda_{k-1}\right)\mathbf{H}_1 + \mathbf{M}_{k-1} \tag{36}$$

which is equivalent to solving the equation :

$$-\left(\Lambda_{k-1}\mathbf{N}+\mathbf{I}\right)\mathbf{M}_{k-1} = \left(\mathbf{I}+\Lambda_{k-1}\right)\mathbf{H}_1. \tag{37}$$

Based on the definition, all the diagonal element of Λ_k satisfy $0 < \lambda_k^{(i)} < 1$, so according to **Lemma 2**, $\left(\Lambda_{k-1}\mathbf{N}+\mathbf{I}\right)$ is invertible. Then $\mathbf{M}_{k-1} = -\left(\Lambda_{k-1}\mathbf{N}+\mathbf{I}\right)^{-1}\left(\mathbf{I}+\Lambda_{k-1}\right)\mathbf{H}_1$ which means such $\mathbf{M}_{k-1}$ that we required exists.

First we perform the following mathematical transformation on Eq. 35:

$$\mathbf{H}_k + \mathbf{M}_{k-1} = -\Lambda_{k-1}\mathbf{N}\left(\mathbf{H}_{k-1}+\mathbf{M}_{k-2}+\mathbf{M}_{k-1}-\mathbf{M}_{k-2}\right), \tag{38}$$

which can be split into the following form:

$$\mathbf{H}_k + \mathbf{M}_{k-1} = -\Lambda_{k-1}\mathbf{N}\left(\mathbf{H}_{k-1}+\mathbf{M}_{k-2}\right) + \left(-\Lambda_{k-1}\mathbf{N}\right)\left(\mathbf{M}_{k-1}-\mathbf{M}_{k-2}\right) \tag{39}$$

Let $\tilde{\mathbf{N}}_{k-1}$ denote $-\Lambda_{k-1}\mathbf{N}$, so the formula can be simply written as:

$$\mathbf{H}_k + \mathbf{M}_{k-1} = \tilde{\mathbf{N}}_{k-1}\left(\mathbf{H}_{k-1}+\mathbf{M}_{k-2}\right) + \tilde{\mathbf{N}}_{k-1}\left(\mathbf{M}_{k-1}-\mathbf{M}_{k-2}\right) \tag{40}$$

We first use Eq. 35 to recurse once, then derive the following formula:

$$\mathbf{H}_k + \mathbf{M}_{k-1} = \tilde{\mathbf{N}}_{k-1}\tilde{\mathbf{N}}_{k-2}\left(\mathbf{H}_{k-2}+\mathbf{M}_{k-3}\right) + \tilde{\mathbf{N}}_{k-1}\left(\mathbf{M}_{k-1}-\mathbf{M}_{k-2}\right) \tag{41}$$

By analogy, continuing to split and iterate, we can get the general term formula of the output of the k-th layer :

$$\mathbf{H}_k = \sum_{i=2}^{k-1}\prod_{j=i}^{k-1}\tilde{\mathbf{N}}_j\left(\mathbf{M}_i-\mathbf{M}_{i-1}\right) + \prod_{i=1}^{k-1}\tilde{\mathbf{N}}_i\left(\mathbf{H}_1+\mathbf{M}_1\right) - \mathbf{M}_{k-1} \tag{42}$$

□

E. Overfitting Experiment

To validate the effectiveness of sampling, we conduct experiments on Cora, Citeseer, and Pubmed using GCN as the base model. Two strategies for learning p are selected: (1) learning directly through backpropagation and (2) first learning the distribution through backpropagation and then sampling. Specifically, we denote the GCN with strategy 1 as L-GCN and the GCN with strategy 2 as SNR-GCN. We set the number of layers to 2, 4, 8, and 16, and test the performance of these two models on the training and validation sets. For each experiment, we run ten times to obtain the mean accuracy along with standard deviation. The results are shown in the Table 1.

Table 1: Node classification accuracy (%) on different number of layers.

Dataset	Method	Type	Layer			
			#2	#4	#8	#16
Cora	L-GCN	Train	99.86±0.28	99.36±0.87	100.00 ±0.00	98.14±2.82
		Validation	79.20±0.63	78.92±0.67	77.86±1.36	77.42±1.99
	SNR-GCN	Train	100.00±0.00	99.79±0.64	98.86±1.81	98.71±2.44
		Validation	80.24±0.47	79.56±0.85	78.02±1.04	78.24±1.20
Citeseer	L-GCN	Train	99.08±1.60	96.00±2.57	93.58±2.93	95.25±2.29
		Validation	70.00±0.84	66.88±1.33	65.68±2.26	66.10±1.31
	SNR-GCN	Train	97.42±2.39	96.33±3.14	96.92±3.09	94.33±4.40
		Validation	70.52±0.56	66.58±1.51	66.80±1.27	66.86±1.16
Pubmed	L-GCN	Train	99.83±0.50	99.83±0.50	99.00±1.10	99.00±1.10
		Validation	78.92±0.64	79.12±0.58	79.10±0.44	79.22±0.67
	SNR-GCN	Train	100.00±0.00	100.00±0.00	99.83±0.50	99.67±0.66
		Validation	79.26±0.22	79.86±0.49	79.42±0.46	79.62±0.46

F. Experiment Setup

F.1 Dataset Statistics

The dataset statics is shown in Table 2. Cora, Citeseer, and Pubmed were applied to all experiments, in addition to using the Chameleon, Squirrel, and CoraFull datasets additionally for the SSNC experiments. For CoraFull, we randomly split all nodes into 40%/10%/50% for training/validation/testing. For Chameleon and Squirrel, we use the same dataset partition as BM-GCN. For the SSNC-MV experiments, we took the same experiments as PairNorm and Group Normalization, removing features from the validation and test sets of Cora, Citeseer, and Pubmed.

Table 2: Dataset Statistics.

	Cora	**Citeseer**	**Pubmed**	**CoraFull**	**Chameleon**	**Squirrel**
#Nodes	2708	3327	19717	19793	2277	5201
#Edges	5429	4732	44338	126842	36101	217073
#Features	1433	3703	500	8710	2325	2089
#Classes	7	6	3	70	5	5
#Training Nodes	140	120	60	7892	1092	2496
#Validation Nodes	500	500	500	1986	729	1664
#Testing Nodes	1000	1000	1000	9915	456	1041

F.2 Parameter Settings

Experiments in Section 3 We apply GCN, GAT with 32 hidden units. We fix the following sets of hyperparameters: dropout = 0.0, weight decay=0.0005. The learning rate is set to 0.01 if the number of layers is less than 16, and 0.001 if the number of layers is greater than or equal to 16.

Experiments in Section 5.2 We apply all the models with 64 hidden units on five datasets. We used the optimal parameters specified in the original papers for APPNP, GCNII and JKNet. For other models, we fix the following sets of hyperparameters: dropout=0.5, weight decay=0.0005, learning rate = 0.01.

Experiments in Section 5.3 We apply all the models with 64 hidden units. We fix the following sets of hyperparameters: dropout=0.5, weight decay=0.0005, learning rate = 0.01. The GCN equipped with the Res module is the GCNII.

Experiments in Section 5.4 We apply all the models with 32 hidden units. We fix the following sets of hyperparameters: dropout=0.5, weight decay=0.0005. The learning rate is set to 0.01 if the number of layers is less than 15, and 0.001 if the number of layers is greater than or equal to 15.

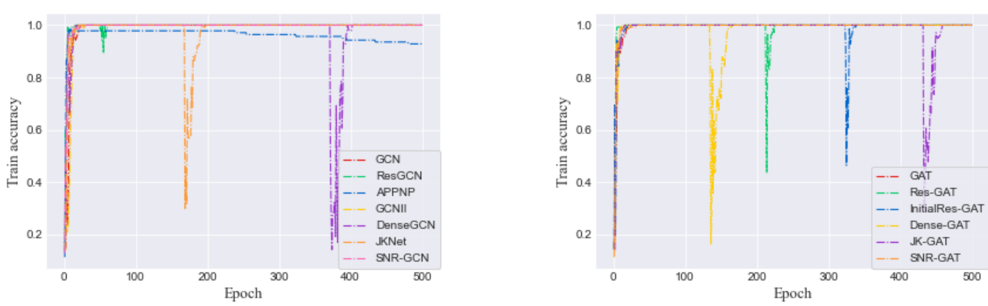
Experiments in Section 5.5 We apply all the models with 64 hidden units. We fix the following sets of hyperparameters: dropout=0.5, weight decay=0.0005. The learning rate is set to 0.01 if the number of layers is less than 16, and 0.001 if the number of layers is greater than or equal to 16.

F.3 Baselines

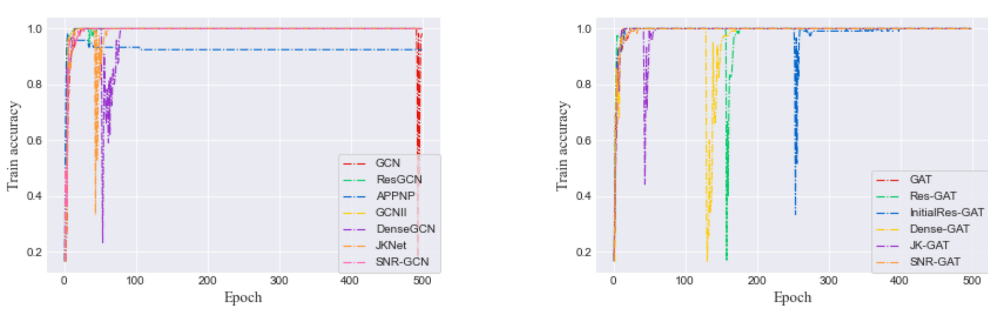
The baseline methods are publicly available at:
- **DGN,PairNorm,BatchNorm:** https://github.com/Kaixiong-Zhou/DGN/
- **DropEdge:**https://github.com/DropEdge/DropEdge
- **Other Models(APPNP,GCNII,...):**https://docs.dgl.ai/en/0.9.x/api/python/nn-pytorch.html

G. SSNC

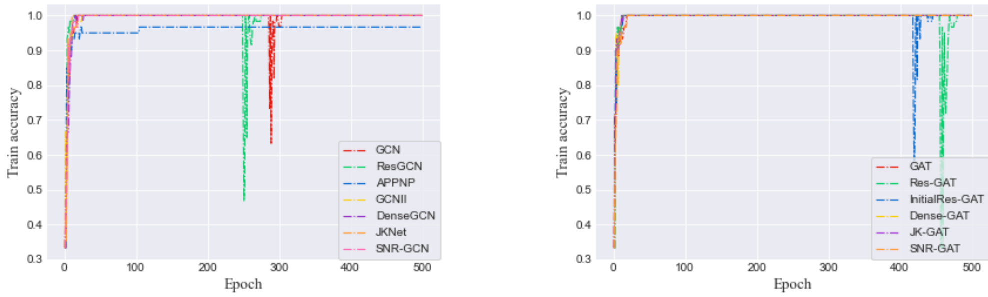
G.1 Train Accuracy



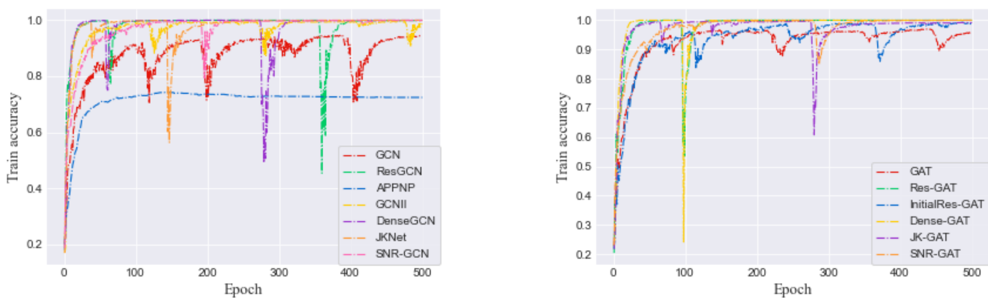
Cora



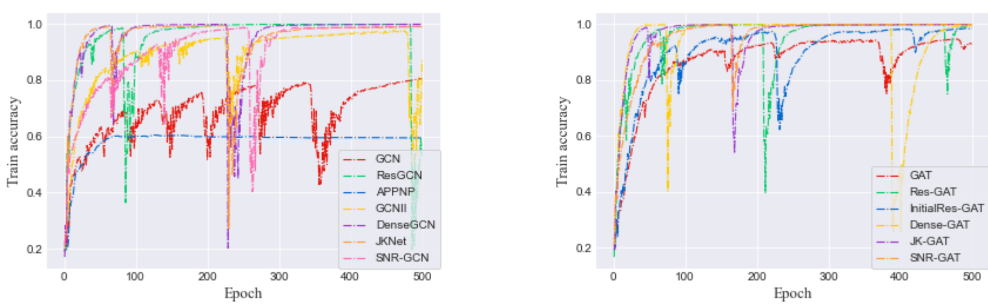
Citeseer



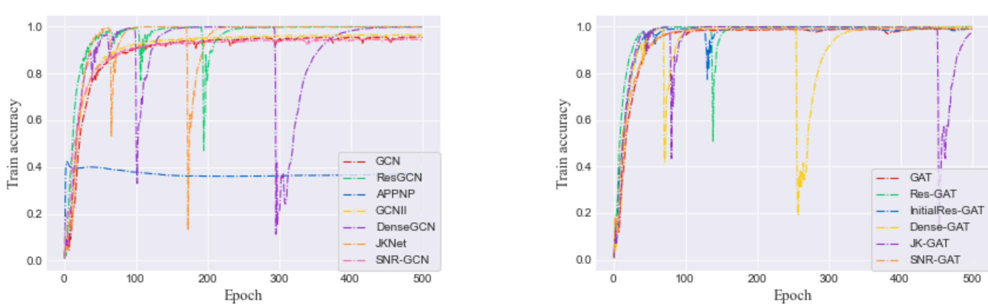
Pubmed



chameleon

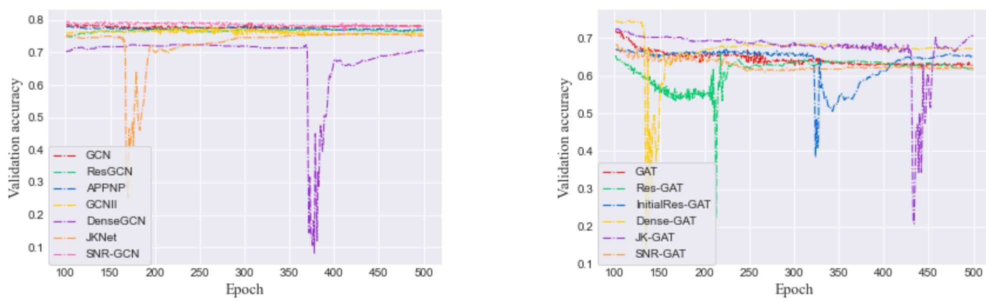


squirrel

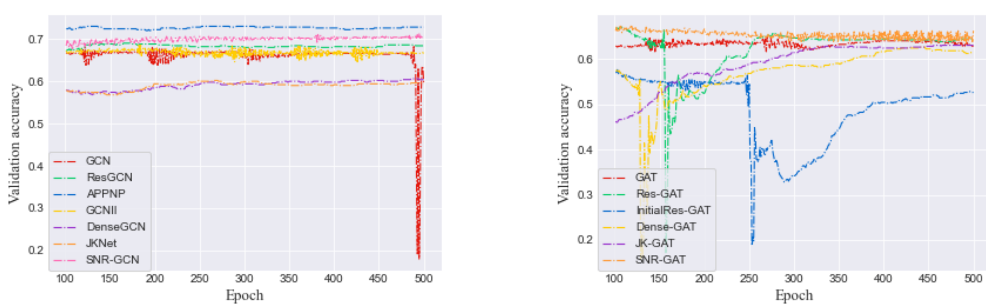


CoraFull

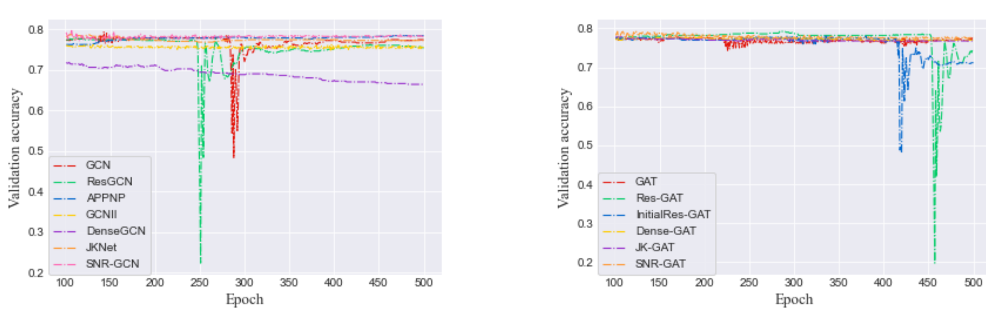
G.2 Validation Accuracy



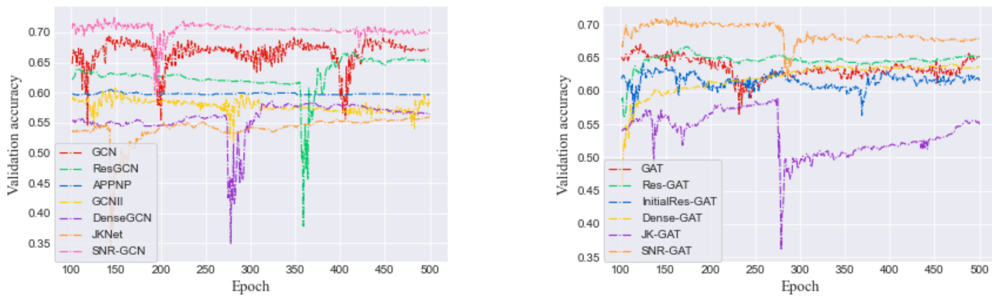
Cora



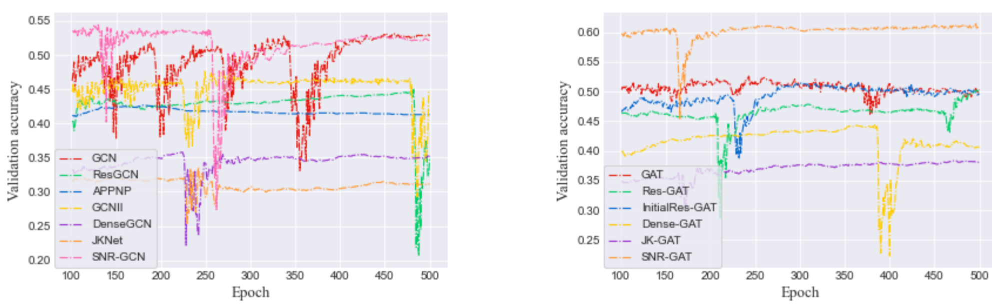
Citeseer



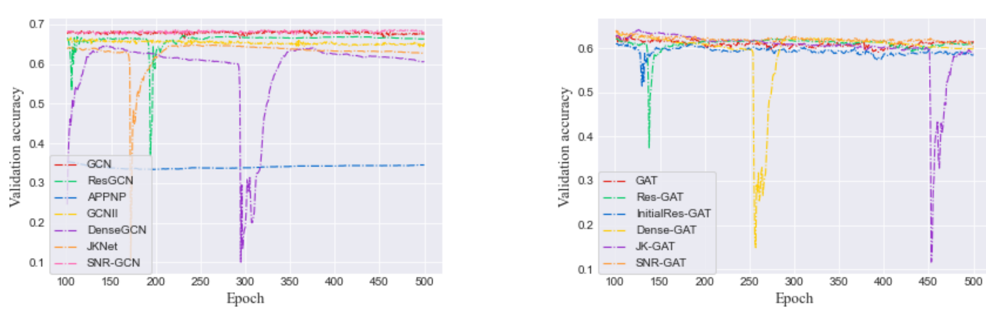
Pubmed



chameleon



squirrel

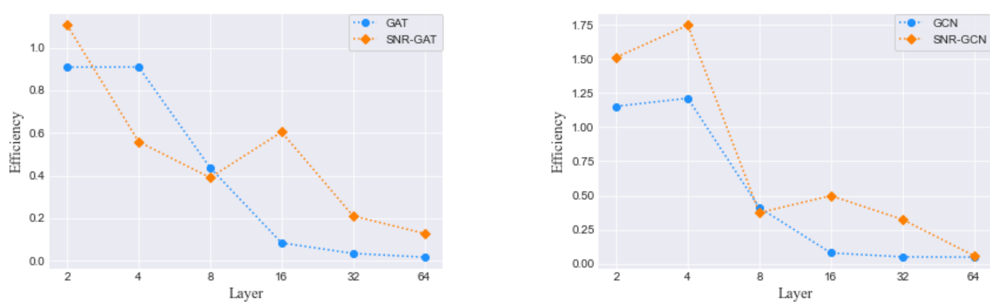


CoraFull

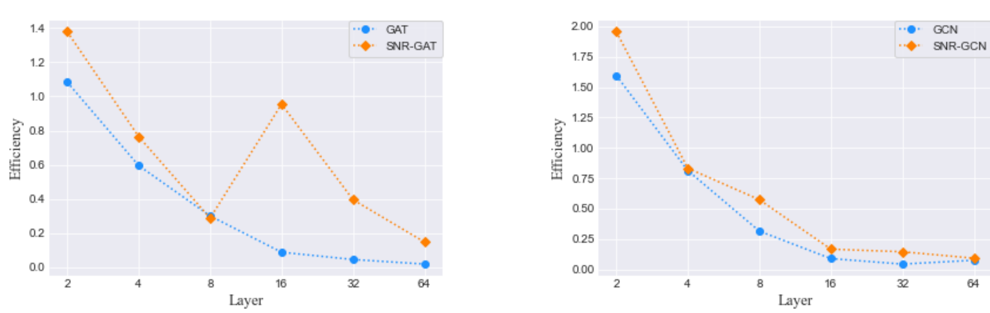
Table 3: Average accuracy of the model over 100 to 500 epoch on the validation set.

Method	Cora	Citeseer	Pubmed	CoraFull	Chameleon	Squirrel
GCN	77.92	66.00	76.73	67.81	66.62	49.12
APPNP	77.37	72.64	77.72	34.07	59.80	41.73
GCNII	76.27	66.83	75.55	65.35	57.80	45.24
ResGCN	76.69	68.32	75.29	66.09	62.86	42.72
JKNet	72.79	58.94	77.38	62.89	54.01	30.97
DenseGCN	68.64	59.12	68.91	58.87	55.84	34.35
Res-GAT	61.32	62.04	77.05	60.90	64.62	46.37
JK-GAT	67.44	58.97	77.01	59.97	53.61	36.92
Dense-GAT	66.65	56.94	77.25	59.64	61.98	41.50
InitialRes-GAT	63.65	48.86	75.88	59.28	61.52	49.23
GAT	64.53	63.36	76.89	60.96	63.45	50.61
SNR-GCN (Ours)	78.58	69.94	78.13	68.14	70.42	51.45
SNR-GAT (Ours)	62.87	65.35	77.84	61.95	68.71	60.19

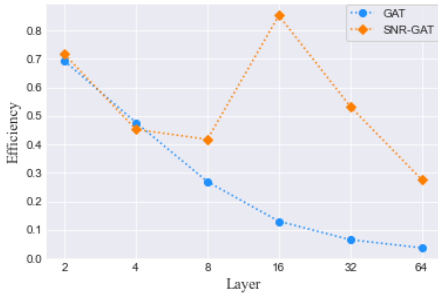
H. Effiency Analysis

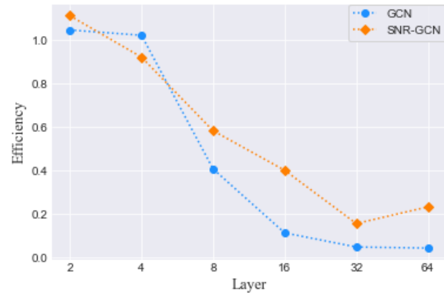


Citeseer



Cora





Pubmed