Q1. *The motivation for the study is unclear. The authors claim that over-smoothing is derived from the changing quality and quantity of neighbors, but this claim lacks strong evidence. It would be helpful if the authors could provide a more detailed explanation and establish connections with previous works, such as [1], to differentiate their approach.*

Reply:

(1) we can define quality and quantity as follows:

Neighbor quality: If two nodes receive the same or similar information from neighbors, we argue that the neighbor quality of the two nodes is low from the perspective of oversmoothing. This can happen to two neighboring nodes in shallow layers as shown in Figure a1 and a2 (nodes A and B). If two nodes have very low-quality neighbors, they will become similar very quickly (oversmoothing happens). According to the small-world phenomenon [17], any two nodes will become neighbors directly or indirectly by 6 jumps. Therefore, with layers increasing, the neighbor quality will become low between any two nodes as shown in Figures a1 and a2 (nodes A and C)

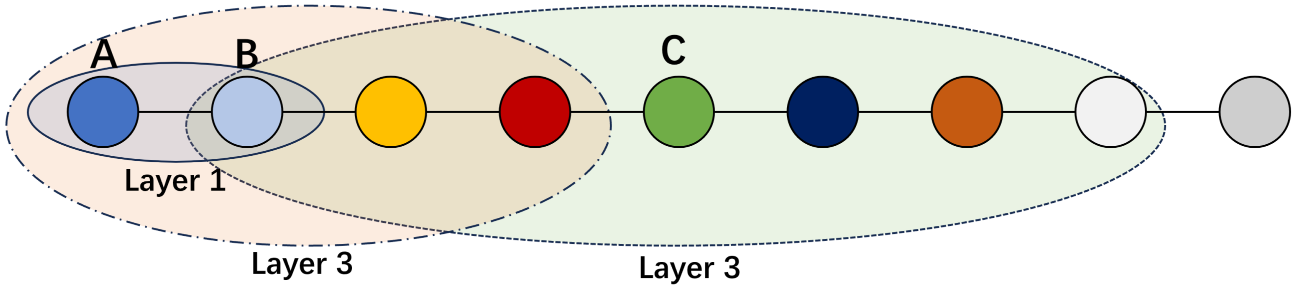


Figure a1. A simplest graph to demonstrate node similarity.

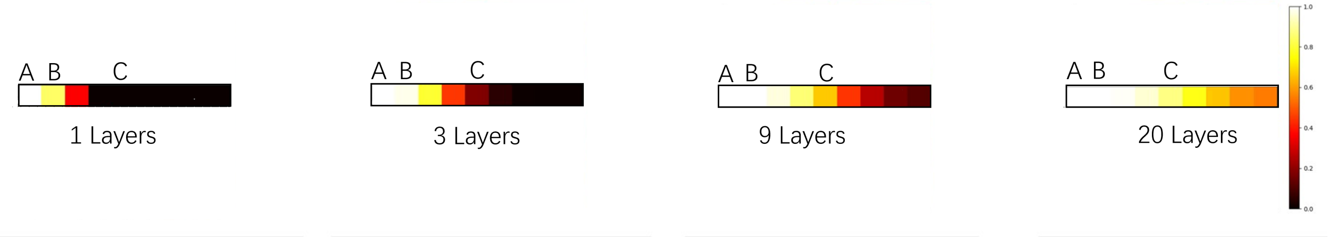


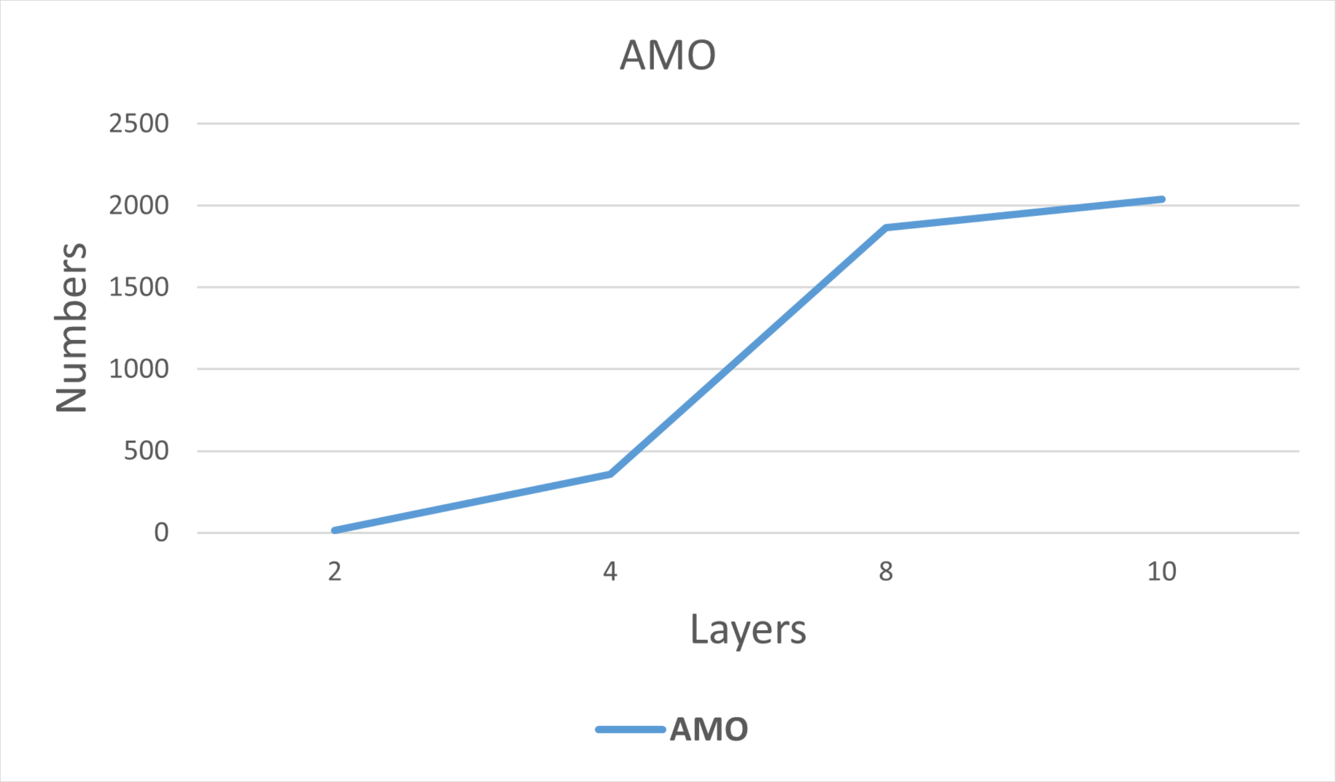
Figure a2. Node similarity changing between A and others. A and B become similar in the 1st layer, because they are neighbor to each other. With layers increasing, A and C becomes similar in layer >3 where they share similar neighbors.

We can use an indicator **Average Mutual Overlapping** (AMO) to describe the neighborhood quality. AMO is defined as follows:

$$S_{i,j} = \begin{cases} 1, A^l_{i,j} > 0 \wedge i \neq j \\ 0, A^l_{i,j} = 0 \end{cases}, Num = \mathrm{Mean}(\mathrm{SS}^T))$$

l is the number of layers and A is the adjacency matrix.

As its name suggests, this indicator averages the number of common neighbors of two nodes over all possible pairs of nodes. It is a sufficient indicator since two nodes have similar aggregation information when they have a large number of common neighbors. On cora, we stack SGC by 2, 4, 8, and 10 layers and measure the AMO. The measurement of the AMO is presented in the Fig. a3.



.

Figure a3. The neighbor quality changeing with respoect to layer numbers

Neighbor quantity: The information received from neighbors grows exponentially, causing nodes to lose their own individuality and accuracy decreases. As shown in Fig. a1, node C has 6 neighbors in layer 3. If the graph become non-linear, the number of neighbors will be exponential to layer number. The Neighbor quantity can be measured by **Average Number of Different Classes of Nodes in the neighbourhood** :

$$ANDCN = \frac{1}{N} \sum_{i=0 j=0}^{N} 1_{[L_i != L_j \wedge S_{i,j}=1]},$$

where N is the number of nodes, L_i the node i's label,$S_{i,j}$ is defined in AMO.

We plot the variation of $ANDCN$ in layers 2, 4, 8 and 10 on the cora dataset in Figure a4.:

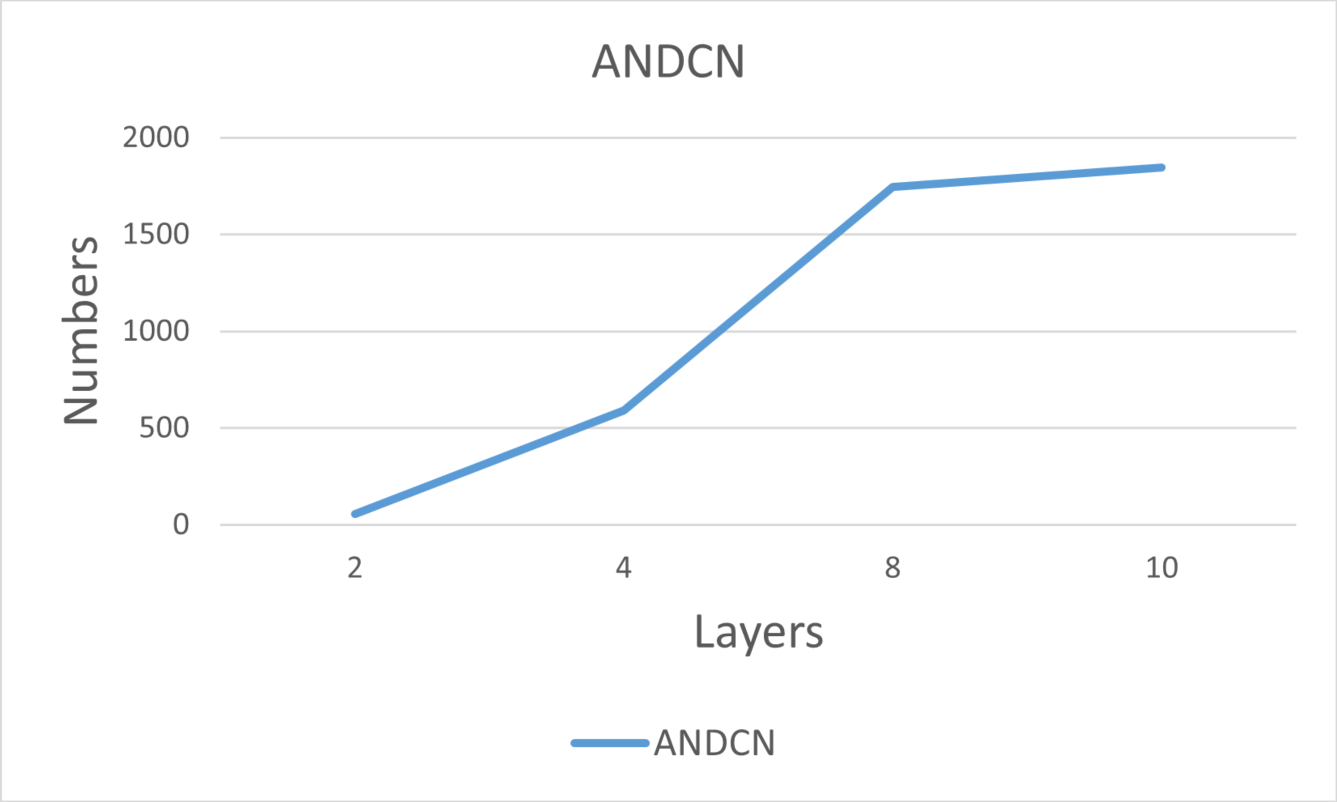


Figure a4. The neighbor quality changeing with respoect to layer numbers

(2) Our motivation can be generally summarized: (a) using random masking to drop a large number of aggregated information and maintain the node's individual information generate from shallow layers. (b) adopting contrastive constraint to enhance node's difference to each other in deep layers.

(3) The provided reference [1] demonstrates that the expressive power of GCNs is related to the topological information. This result is also can be demonstrated in Figure 2a. If two nodes are close neighbors they share neighbors and have low-quality neighbors. As a result, A and B lose their express power in very shallow layers. However A and C don't have this kind of topology, they lose expressive power after layer 9.

Q2 *The random masking module in TSC appears similar to the "column" version of SkipNode [2], yet the authors have not cited SkipNode or discussed the differences between the two. Both approaches involve replacing representations of the current and previous layers. It would be valuable for the authors to address this similarity and highlight the distinctions between TSC and SkipNode.*

Reply:

Figure a5 shows the difference between GCN, TSC, and SkopNode. From the matrix alone, SkipNode and TSC seem to have only rows and columns difference; In the SkipNode, one node may be note update in deep layers, if we set a large drop rate. TSC requires all nodes to be updated in sub-columns, which can guarantee each node receives information. The feature updating strategy can guarantee all nodes receive all layers' neighbor information in a slowing manner. Moreover, TSC adds a contrast constraint to increase the variability among nodes, which further prevents over-smoothing.

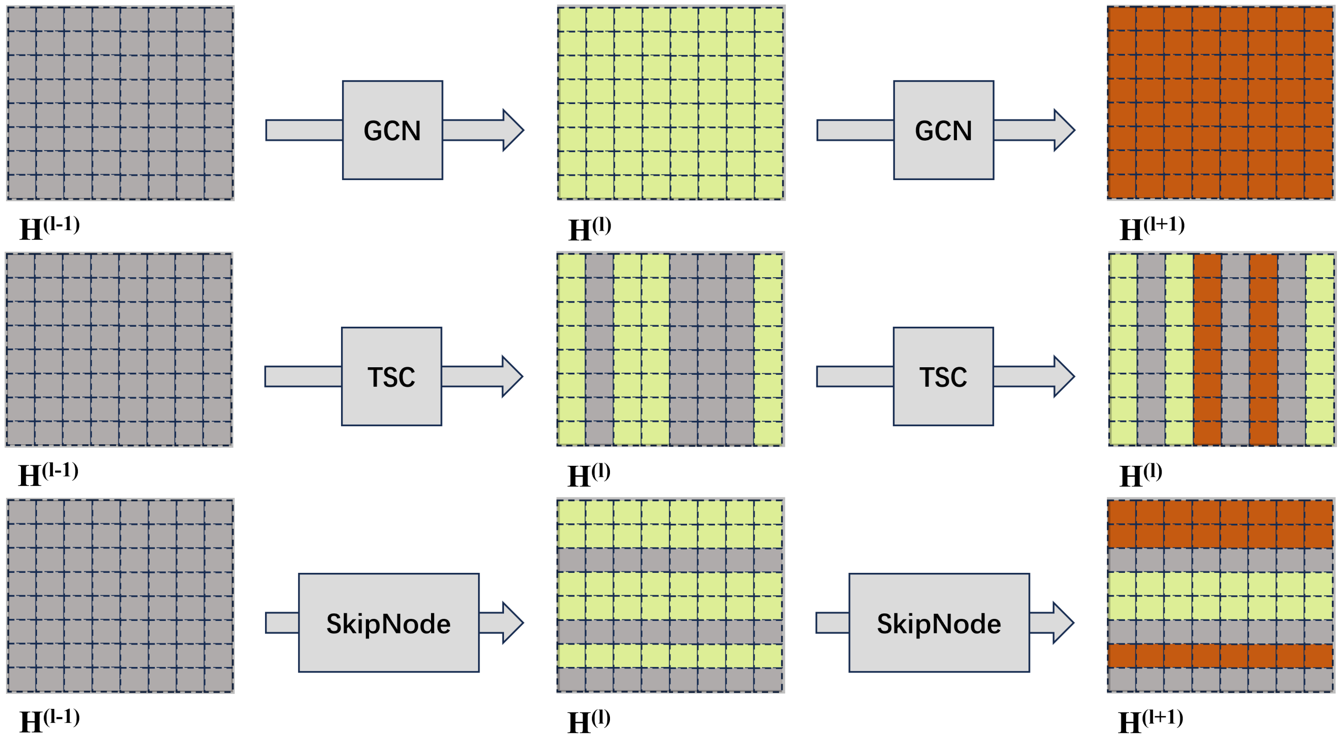


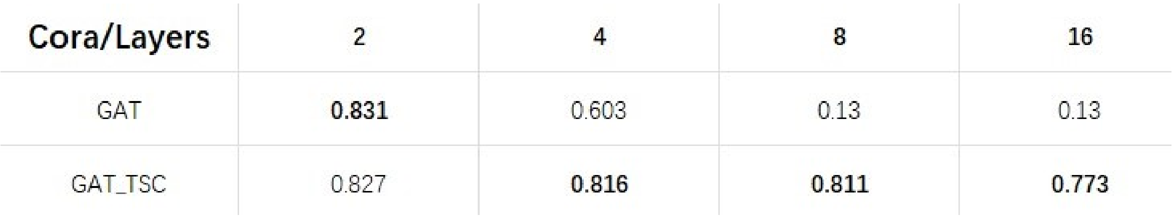
Figure a5. The difference among TSC, GCN and SkipNode.

Q3 *While the authors claim that TSC is a plugin module, the theoretical analysis is limited to SGC. Furthermore, the experimental section only demonstrates the results of applying TSC to GCN and SGC, which are simple and representative models. It would be beneficial to investigate the effectiveness of TSC on more advanced GCN models, such as GCNII [3], APPNP [4], GAT [5], and GRAND [6].*

Reply:

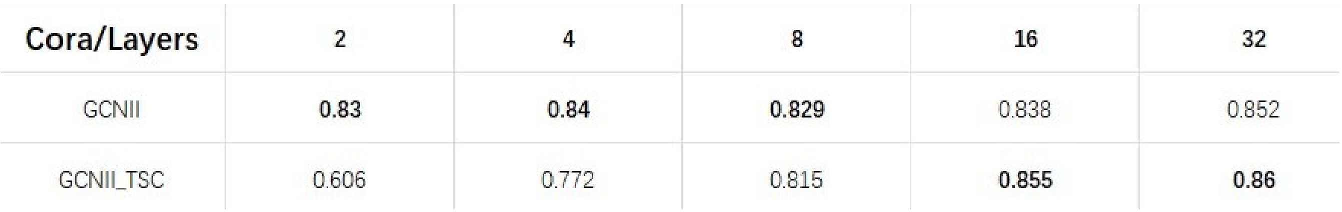
We applied TSC technique to GAT and GCNII with the following effect, the code has been uploaded. The results on Cora datset are reported in Table a1 and a2.

Table a1. ACC on GAT and GAT+TSC.



Cora/Layers	2	4	8	16
GAT	**0.831**	0.603	0.13	0.13
GAT_TSC	0.827	**0.816**	**0.811**	**0.773**

Table a2. ACC on GCNII and GCNII+TSC.



Cora/Layers	2	4	8	16	32
GCNII	**0.83**	**0.84**	**0.829**	0.838	0.852
GCNII_TSC	0.606	0.772	0.815	**0.855**	**0.86**

From Table a1 and a2, it can be seen that apply TSC to GAT and GCNII can boost the performance in Cora dataset.

Q4 Some important *related works, such as [1, 2, 4, 6], are missing from the paper's references. Including these works would provide a more comprehensive overview of the research landscape and strengthen the paper's credibility.*

Reply: Thanks for you kind questions. We will include them in revisions.

Q5 *In line 459, "w" should be capitalized as "W."*

Reply: Thanks for you kind questions. We will fix it in revisions.

Refs:

[a1] GRAPH NEURAL NETWORKS EXPONENTIALLY LOSE EXPRESSIVE POWER FOR NODE CLASSIFICATION, ICLR 2020

[a2] SkipNode: On Alleviating Performance Degradation for Deep Graph Convolutional Networks, Arxiv

[a3] Simple and deep graph convolutional networks, ICML 2020

[a4] Predict then propagate: Graph neural networks meet personalized pagerank, ICLR 2019

[a5] Graph attention networks, ICLR 2018

[a6] Graph random neural networks for semi-supervised learning on graphs, NIPS 2021