Mutual Concerns of All Reviewers

"*The results around Fig. 1 are difficult to understand; Some figure and table captions and some mathematical derivations should be revised; The experiment section needs to be expanded; Future works in this current form is not very useful.*": We will rewrite the corresponding sections and fix the issues you have pointed out. Thank you all very much!

Mutual Concerns of Reviewer 2 and Reviewer 3

"*What factors are critical for the performance of the proposed architectures? Ablation tests to see either the network architecture or the activation works is needed.*": The changes in network architecture and in the activation function both contribute to the "scalability" of the network: the ability of increasing the expressive power by enlarging the network. Empirically, such "scalability" is observed for that larger instances of the 2 architectures yield better performance. We have done ablation tests with extensive tuning in each architecture to see their limits. For those best configurations, we observed that the change in activation or architecture alone both contributes to the performance: the architecture contribution is more significant with fewer training labels while that of activation function is the opposite. However, when combined, they did not result in '1+1=2': the contribution of the activation seems being absorbed. Such observation is expected since the number of layers does not suffice to demonstrate significant performance difference also for the characteristics of the tasks. Larger difference with more layers is expected on more complex tasks.

Mutual Concerns of Reviewer 3 and Reviewer 4

"*Clarify the motivation of the proposed architectures and the necessity of the two different architectures.*":

The oversmoothing problem from which classical GCN suffers, when adding more layers, can be intuitively interpreted that low-level information is neglected in the higher level of information diffusion, as there are no direct connections in the architecture. The two proposed architectures address this issue by stacking levels of information together in **different** manners: Snowball accumulatively stacks those features layer by layer, whereas Truncated Krylov considers all levels of diffusion information simultaneously in each layer.

Concerns of Reviewer 2 Only

"*More experiments on different tasks, provide a complexity (time and memory) analysis.*": Results on larger datasets and inductive learning tasks will be added, with more results to illustrate the arguments about the activation. The complexity (time and memory) of the new architectures is clearly larger than GCN's due to the dense connected nature of Snowball architecture and the size of the Truncated Krylov networks. As the complexity depends on the sparsity of the matrices (task-dependent) and the mechanisms of pytorch, it is hard to analyze it theoretically. However, we will provide details about the runtime and memory consumptions in the experimental section.

Concerns of Reviewer 4 Only

"*The authors refer to scalability issues for GCN in the sense of stacking multiple layers but the term refers to scalability wrt size of the input.*": We will state more clearly, *e.g.* the scalability of the size of the network.

"*Why is the graph defined using edges and adjacency? Isn't it enough to have either one?*": We will fix this.

"*Chebyshev polynomial constitutes a spectrum-free. The method does not require the computation of the eigendecomposition, however the resulting method still behaves as spectrum-based.*": Our original statements aligned with the naive dichotomy of some existing work, where spectrum-free refers also to those behave as spectrum-based with no explicit eigendecomposition. But we do think that your dichotomy is more reasonable. We will make the change.

"*How does the method compare to approaches based on the more general message passing paradigm that can implement both local and global computation? Laplacian smoothing is not necessarily an issue there.*": Denote $N^k(v)$ as the $k-$hop neighbors of node v and $\|$ as concatenation. Message passing paradigm cannot avoid oversmoothing because it does not leverage multi-scale information in each layer. In fact, we need a densely connected architecture. The relations are illustrated in the following table. We can also change the readout function $\hat{y} = R(\{h_v^T, |v \in G\})$ to $\hat{y} = R(\{h_v^0, h_v^1, \ldots, h_v^T, |v \in G\})$ to mitigate oversmoothing.

	Massage Passing	GraphSAGE-GCN	Snowball	Truncated Krylov
Matrix	$m^{t+1} = M_t(A, h^t)$	$m^{t+1} = Lh^t$	$m^{t+1} = L[h^0\| \ldots \|h^t]$	$m^{t+1} = h^t\| \ldots \|L^{m_t - 1}h^t$
	$h^{t+1} = U_t(h^t, m^{t+1})$	$h^{t+1} = \sigma(m^{t+1}W^t)$	$h_v^{t+1} = \sigma(m^{t+1}W^t)$	$h^{t+1} = \sigma(m^{t+1}W^t)$
Nodewise	$m_v^{t+1} = \sum\limits_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw})$	$m_v^{t+1} = \text{mean}(\{h_v^t\} \cup \{h_{N(v)}^t\})$	$m_v^{t+1} = \|_{i=0}^{t}\text{mean}(\{h_{N(v)}^i\})$	$m_v^{t+1} = \|_{i=0}^{m_t - 1}\text{mean}(\cup_{k=0}^{i}\{h_{N^k(v)}^t\})$
	$h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$	$h_v^{t+1} = \sigma(W^t m_v^{t+1})$	$h_v^{t+1} = \sigma(W^t m_v^{t+1})$	$h_v^{t+1} = \sigma(W^t m_v^{t+1})$