We thank all reviewers for the constructive comments and are glad to see overall positive reviews toward this work.

Response to R1: Thank you for acknowledging contributions of our work and offering helpful suggestions.

***Weakness (#1)*:** The symmetric trick is an intuitive way, while our Eq.(1) is more interpretable and can degenerate to the trick form. Moreover, as mentioned in Line 91-94, our method uses eigenvector matrix to normalize Laplacian which considers nodes' in & out-connection respectively, while the symmetric trick sums in & out -degree matrix together to normalize Laplacian. It ignores the directed structure and may lead to degenerated performance. ***(#2)*:** Exactly, intersection means having both meeting and diffusion paths. And according to the definition in Line 162, when $k = 1$, we cannot find zero length paths. Thus, transition matrix is used to find directly connected neighbors.

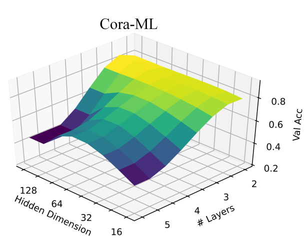
Suggestion: Please refer to ***Space Complexity*** under **R3**. Sorry about merging answers due to page limit. Meanwhile, we will update the paper to discuss high-order GCN ideas, revise the typos and make figures explanation clearly. We will also provide baseline results using simple symmetric adjacency matrix in the Supp. Thank you.

Response to R3: Thank you for invaluable feedback. ***Model Selection*:** We apologize that we did not describe all details of model selection and raised misleading of choosing test/val set due to page limits. For the digraph convolution models without Inception block, we set the network architectures and hyperparameters same with GCN since they used similar spectral analysis. For the model with IB, as we mentioned in Line 244-246, we did split the validation set and carried out the grid search on it. The val accuracy on CORA-ML and AM-COMPUTER with the # of layers and hidden dimension are shown in the Figure (a,b) respectively. We chose a three layer model and set it hidden dimension to 32. Our model trended similarly on the val & test sets, causing the misleading that we selected models on the test set. We also did grid search on the hyperparameters: lr in range [0.001, 0.1], weight decay in range [1e-5, 1e-3] and dropout rate in range [0.3,0.7]. The network architectures and hyperparameters were presented in the **Supp Section 4** and we will update the paper with detailed model selection process. ***Space Complexity***: Due to the asymmetry of the digraphs mentioned in Line 151-154, long paths normally exist between a few points and are not bidirectional. Thus, using k^{th}-order proximity will get unbalanced receptive field and introduce computational complexity. Intersection and union of meeting and diffusion paths both can handle unbalancing problem. We compare the # of edges per Inception block using intersection and union with different k on two datasets shown in Figure (c) and find that intersection does help to reduce the memory-consuming. Thus, we report the practical case that space complexity grows linearly with the # of edges empirically. However, the worst case does exist when the input graph is undirected and strongly connected. Though it is unsuitable to our model, which mainly treats reducible digraphs, we will modify the space complexity to $\mathcal{O}(k|\mathcal{V}|^2)$ and add detailed explanations. We will also revise the paper following your remarks. Thank you.

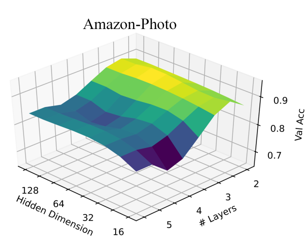
Response to R5: We appreciate the detailed and positive comment, which reflects essential contributions of our work.

***Weakness (#1)*:** We fully agree that additional experiments are needed. We will update the article to add link prediction task and compare with listed papers. ***(#2)*:** The OOM model is our initial model based on PageRank indeed. We realize this (see Line 101-104) and further simplify it using *personalized* PageRank. In Table 2, our simplified method not only solves the OOM problem, but also improves the accuracy. We also plot training time per epoch on random digraphs with difference size (with N nodes and $2N$ edges) in Figure (d) to show you that our model can handle about 10 million nodes in one GPU (11GB). Moreover, we will add detailed experiments on large datasets in the Supp. Thank you.

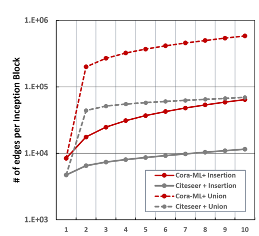
Response to R6: We thank you for giving considerate review and the opportunity to explain. ***Weakness (#1)*:** Our second contribution is proposing Inception module in digraphs to enable a larger receptive field, while k^{th}-order proximity is a part of it. Digraphs have unique features mentioned in Line 151-154 and neither the existing k-hop nor the k^{th}-order method defined on undirected graphs can handle them well. Therefore, we redefine k^{th}-order proximity using intersection in digraphs. It can achieve high performance while reduce the computational complexity. ***(#2)*:** We used the source code from DGCN's authors in our experiments. The differences in the results are due to two main factors. First, the experimental tasks in the two papers are different. Although DGCN uses digraph datasets, their inputs are still symmetric adjacency matrices, which is why baselines in DGCN are similar to the original results in undirected graphs. Our experiments, however, restrict the inputs to asymmetric adjacency matrices in order to measure their performance in digraphs. Please see Line 227-233 for experimental task and Line 254-257 for analysis. Second, DGCN concatenates second-order in & out matrix to obtain neighbor features, which could lead to a significant increase in the # of edges and cause OOM problem. Our approach goes beyond DGCN not only for better performance under more stringent experimental conditions, but also for better interpretability. DGCN uses simple symmetric matrices for first-order proximity in its Eq.(8) and does not explain why directed structural features can be obtained. Thank you.



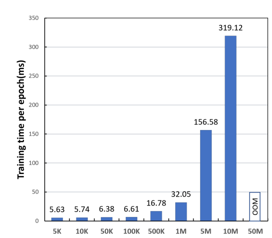
(a)



(b)



(c)



(d)