

1 **Reviewer #1:** Thank you for the positive comments and suggestions! Below we address your questions in detail.  
 2 *R1Q1. It would be better if authors can try dropedge and sampling methods, instead of only adopting dropnode.*

3 We have to emphasize that GRAND is a general framework, and random propagation can be achieved through various sampling methods, including DropNode, dropout, DropEdge or other methods. Indeed, besides DropNode, we also tried dropout as augmentation method in paper. In the past of few days, we have tried to implement another variant—GRAND\_DropEdge, which adopts DropEdge as perturbation method. Specifically, we randomly remove some elements from adjacency matrix and use the perturbed matrix to perform mix-order propagation. Table 6 shows the classification results on benchmarks. we run 100 random trials for each dataset. As we can see, though GRAND\_DropEdge gets worse performance than GRAND (DropNode) and GRAND\_dropout, it still outperforms most of baselines (Cf. Table 1).

Table 6: Results of GRAND\_DropEdge (%).

	Cora	Citeseer	Pubmed
GRAND_DropEdge	84.5 ± 0.3	74.4 ± 0.4	80.9 ± 0.9

12 *R1Q2. It would be better if authors can provide the performance under different training ratio.*  
 13 We have conducted experiments to evaluate GRAND under different label rates. For each label rate setting, we randomly create 10 data splits, and run 10 trials with random initialization for each split. We compare GRAND with GCN and GAT. The results are shown in Table 7. We observe that GRAND consistently outperforms GCN and GAT across all label rates on three benchmarks.

Table 7: Classification Accuracy under different label rates (%).

Dataset	Cora					Citeseer					Pubmed				
Label Rate	1%	2%	3%	4%	5%	1%	2%	3%	4%	5%	0.1%	0.2%	0.3%	0.4%	0.5%
GCN	62.8±5.3	71.5±2.3	76.1±1.9	78.6±1.6	79.6±2.1	63.4±2.9	68.3±2.1	70.6±1.7	71.5±1.3	72.2±1.1	71.5±2.1	75.5±1.9	77.5±1.8	79.3±1.7	80.8±1.5
GAT	64.3±5.8	73.5±2.8	77.2±2.4	79.9±2.0	80.8±2.1	64.4±2.9	68.6±1.9	70.4±1.9	71.4±1.3	72.0±1.3	72.0±2.1	75.7±2.0	77.6±1.6	79.3±1.7	80.6±1.2
GRAND	69.1±4.0	76.7±2.5	79.5±2.2	82.7±1.6	83.0±1.6	65.3±3.3	70.0±2.0	72.3±1.8	73.0±1.1	73.8±0.9	74.7±3.4	79.9±2.5	81.4±2.1	82.7±1.6	83.8±1.3

16  
 17 *R1Q3. The proposed methods are not that novel.*

18 Though some related techniques have been studied in previous works, our work is not a trivial combination of them. Our contributions lie in the general framework for graph-based semi-supervised learning. The proposed method is simple yet effective and beats 14 state-of-the-art baselines on standard benchmarks. More importantly, the framework has a well-established theoretical guarantee. To the best of our knowledge, this is the first work that provides theoretical explanations for applying consistency regularization on graph data.

23 *R1Q4. It would be better if the authors can try Stanford OGB ... add GraphSAINT & LADIES*  
 24 We’ve reported the results of 14 GNN baselines on widely-used datasets—Cora, Citeseer, and Pubmed, as well as the performance of 6 more datasets in Appendix. Due to the limited response time, we prioritize experiments for Q1 and Q2. We will have results on OGB and discuss and compare LADIES and GraphSAINT in next version.

27 **Reviewer #2:** Thank you for the positive comments! We address your specific concerns in detail below.

28 *R2Q1. The technical novelty of DropNode needs further discussion, which is similar to FastGCN/GraphSAGE.*  
 29 Since GRAND is a general framework, the sampling methods used in FastGCN/GraphSAGE can also be used in GRAND for graph data augmentation. Here we’d like to talk more about the differences between these two kinds of sampling methods. In terms of the objective, FastGCN/GraphSAGE is mainly used for scaling and accelerating GCNs, while DropNode is aimed to perform graph data augmentation to improve model’s generalization capacity. Technically, GraphSAGE adopts node-wise sampling, which is much less efficient than DropNode as it requires recursive sampling of neighborhoods for every node. FastGCN uses layer-wise importance sampling, in which the sampling probability is related to node’s degree. Compared to that, DropNode samples each node based on an i.i.d. Bernoulli distribution, which will generate more stochastic augmentations and thus works better than importance sampling in improving model’s generalization. Overall, the sampling methods used in FastGCN/GraphSAGE are not optimal options for graph data augmentation when compared with DropNode. One promising research direction is to combine these two kinds of sampling methods to make GRAND more scalable. We will leave this part into our future work.

40 *R2Q2. Why DropNode can bring significant performance gains while DropEdge cannot?*  
 41 As we claimed in paper (line 235 & line 229), the results of DropEdge reported in Table 1 are taken from its original paper [29], wherein GCN serves as the backbone model. To make narration more clear, here we rename it GCN\_DropEdge. GCN\_DropEdge cannot get as much performance improvement as GRAND for two reasons. First, GCN\_DropEdge does not use consistency regularization, the critical strategy for improving model’s generalization used in GRAND. Second, in GCN\_DropEdge, feature propagation is coupled with nonlinear transformation, leading to oversmoothing and overfitting issues when increasing model’s layer num. This causes GCN\_DropEdge cannot perform high-order message passing like GRAND. In recent days, we have incorporated DropEdge into GRAND framework, named GRAND\_DropEdge (Cf. the response to *R1Q1*). GRAND\_DropEdge does not have the above limitations and outperforms GCN\_DropEdge by a large margin (Cf. Table 6).

50 **Reviewer #4:** Thank you for the encouraging comments! We have provided the classification results under different label rates in Table 7. We will provide more analysis on the limitations of GRAND in the next version.  
 51