Graph Random Neural Networks for Semi-Supervised Learning on Graphs

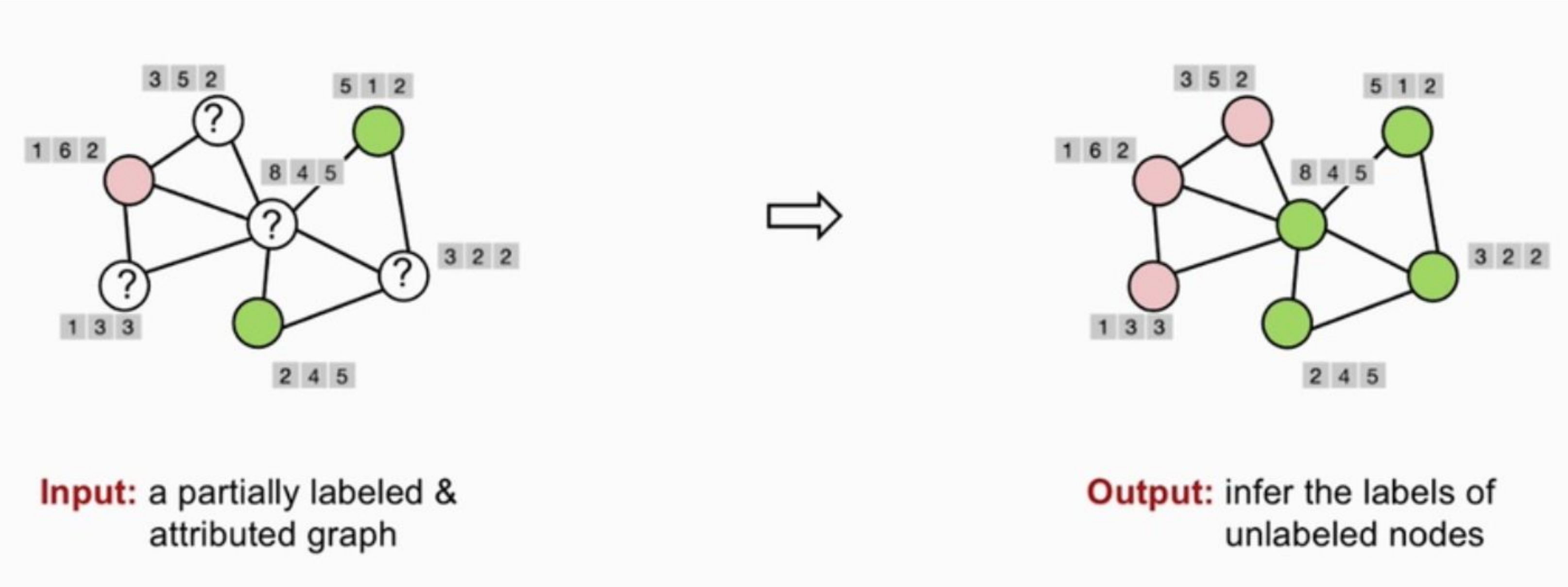
Presented by:
Gauransh Sawhney 2018A3PS0325P
Utkarsh Kumar Singh 2018A3PS0368P

In partial fulfillment of the requirements of the course:
BITS F464 Machine Learning
Submitted to: Dr. Kamlesh Tiwari

Semi-Supervised Learning

- What is Semi-Supervised Learning
- Semi-Supervised Learning on Graphs

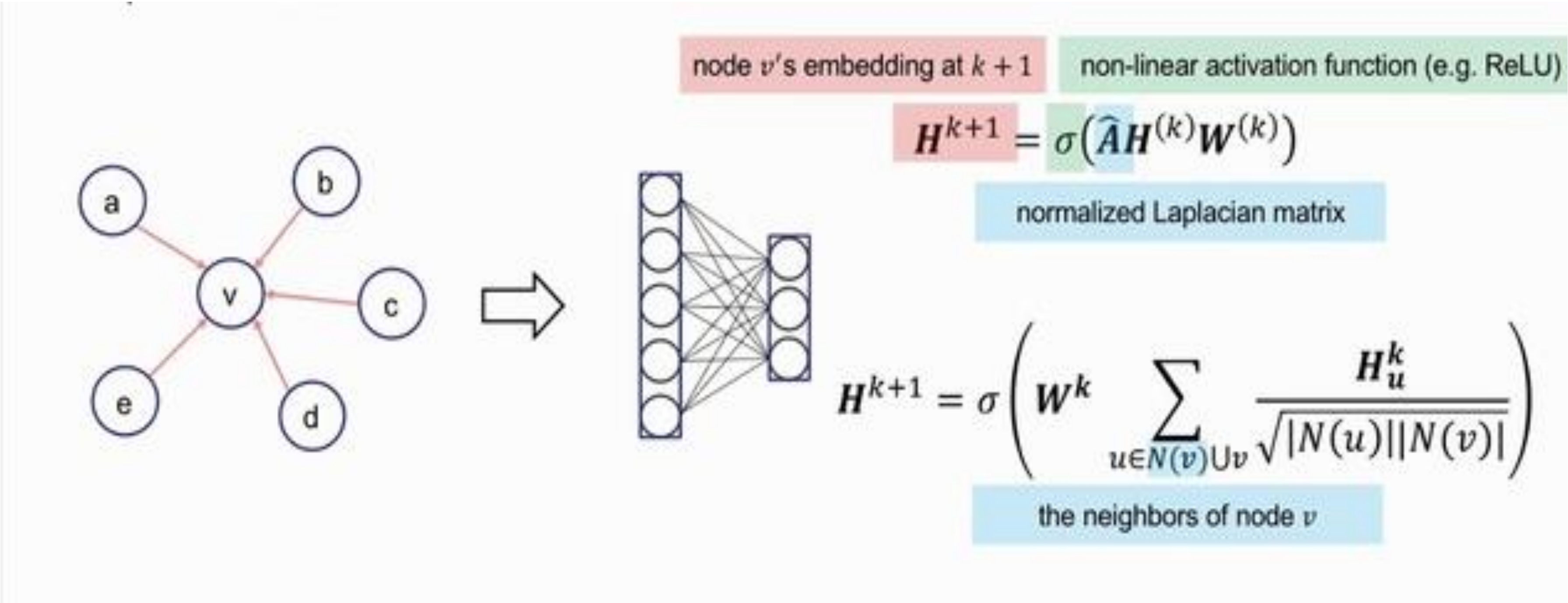
Semi-Supervised Learning on Graphs



Graph Neural Networks

- What are Graph Neural Networks(GNNs)
- Problems with existing implementations

Graph Neural Network



Existing Issues

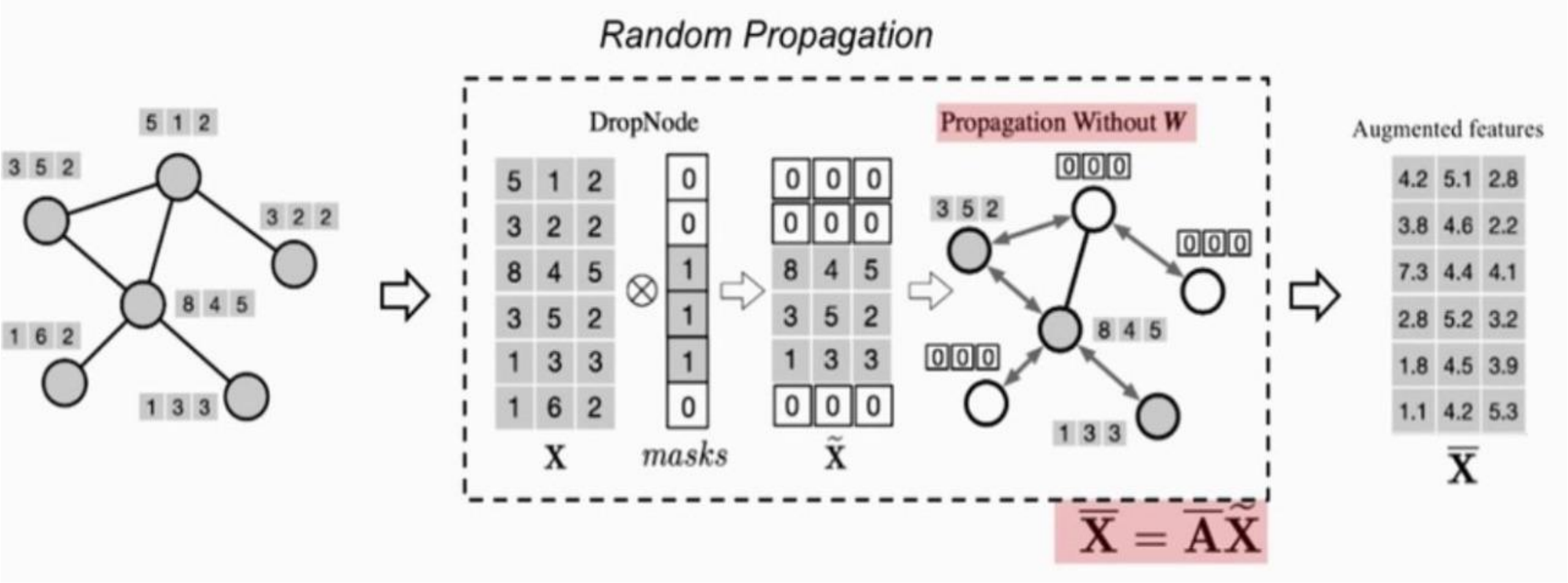
$$H^{k+1} = \sigma\left(\hat{A}H^{(k)}W^{(k)}\right)$$

- Oversmoothing
  - Stacking multiple GNN layers makes nodes indistinguishable; coupling the feature propagation and non-linear transformation steps, aggravates this problem
- Not robust to graph attacks
  - Each node is highly dependent on neighbors, making it non-robust to noise
- Overfitting in case of semi-supervised
  - In the standard setting of semi-supervised training, scarce node label information can be overfit

GRAND: Graph Random Neural Network

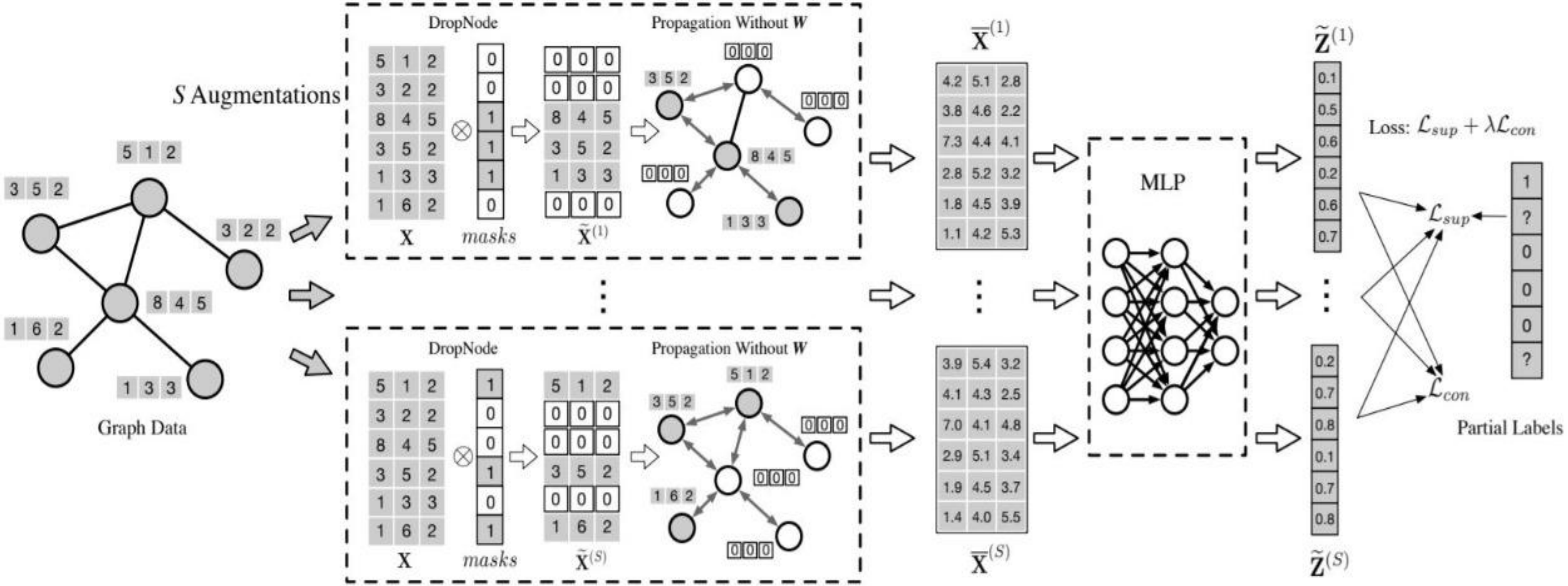
- Architecture
- Algorithm
- How does it tackle the issues faced by other GNNs

GRAND



Random Propagation
5 1 2
3 5 2
3 2 2
8 4 5
1 6 2
1 3 3
DropNode
X
masks
$\tilde{X}$
Propagation Without W
0 0 0
3 5 2
8 4 5
1 3 3
Augmented features
4.2 5.1 2.8
3.8 4.6 2.2
7.3 4.4 4.1
2.8 5.2 3.2
1.8 4.5 3.9
1.1 4.2 5.3
$\overline{X}$
$\overline{X} = \overline{A}\tilde{X}$

GRAND



S Augmentations
DropNode
Propagation Without W
X
masks
$\widetilde{X}^{(1)}$
$\widetilde{X}^{(S)}$
$\overline{X}^{(1)}$
$\overline{X}^{(S)}$
MLP
$\widetilde{Z}^{(1)}$
$\widetilde{Z}^{(S)}$
Loss: $\mathcal{L}_{sup} + \lambda\mathcal{L}_{con}$
$\mathcal{L}_{sup}$
$\mathcal{L}_{con}$
Graph Data
Partial Labels

Algorithm

Algorithm 1 GRAND

Input:

Adjacency matrix $\hat{\mathbf{A}}$, feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, times of augmentations in each epoch S, DropNode/dropout probability δ, learning rate η, an MLP model: $f_{mlp}(\mathbf{X}, \Theta)$.

Output:

Prediction $\mathbf{Z}$.

while not convergence **do**
    for $s = 1 : S$ **do**
        Pertube the input: $\widetilde{\mathbf{X}}^{(s)} \sim \text{DropNode}(\mathbf{X}, \delta)$.
        Perform propagation: $\overline{\mathbf{X}}^{(s)} = \frac{1}{K+1} \sum_{k=0}^{K} \hat{\mathbf{A}}^k \widetilde{\mathbf{X}}^{(s)}$.
        Predict class distribution using MLP: $\widetilde{\mathbf{Z}}^{(s)} = f_{mlp}(\overline{\mathbf{X}}^{(s)}, \Theta)$
    end for
    Compute supervised classification loss $\mathcal{L}_{sup}$ via Eq. 1 and consistency regularization loss via Eq. 3.
    Update the parameters Θ by gradients descending: $\Theta = \Theta - \eta \nabla_\Theta (\mathcal{L}_{sup} + \lambda \mathcal{L}_{con})$
end while
Output prediction $\mathbf{Z}$ via: $\mathbf{Z} = f_{mlp}(\frac{1}{K+1} \sum_{k=0}^{K} \hat{\mathbf{A}}^k \mathbf{X}, \Theta)$.

Loss Functions

$$\mathcal{L}_{sup} = -\frac{1}{S}\sum_{s=1}^{S}\sum_{i=0}^{m-1}\mathbf{Y}_i^{\top}\log\widetilde{\mathbf{Z}}_i^{(s)}.$$

$$\overline{\mathbf{Z}}_{ij}^{'} = \overline{\mathbf{Z}}_{ij}^{\frac{1}{T}} \Bigg/ \sum_{c=0}^{C-1}\overline{\mathbf{Z}}_{ic}^{\frac{1}{T}}, (0 \leq j \leq C-1),$$

$$\mathcal{L}_{con} = \frac{1}{S}\sum_{s=1}^{S}\sum_{i=0}^{n-1}\|\overline{\mathbf{Z}}_i^{'} - \widetilde{\mathbf{Z}}_i^{(s)}\|_2^2.$$

Results

Some of the results presented in the paper:

- Comparison with existing architectures on benchmarks
- Generalization analysis
- Robustness analysis
- Over-smoothing analysis
- Results on large datasets

Dataset Description

3 datasets were used to benchmark results

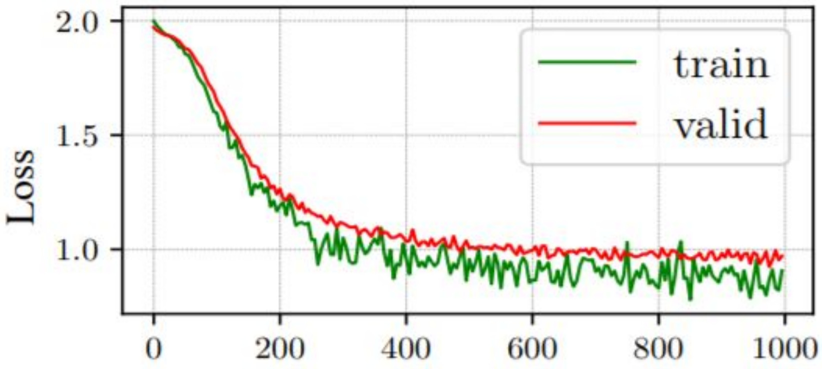
Dataset	Nodes	Edges	Train/Valid/Test Nodes	Classes	Features	Default Label Rate
Cora	2708	5429	140/500/1000	7	1433	0.052
Citeseer	3327	4732	120/500/1000	6	3703	0.036
Pubmed	19717	44338	60/500/1000	3	500	0.003

Comparison with existing architectures

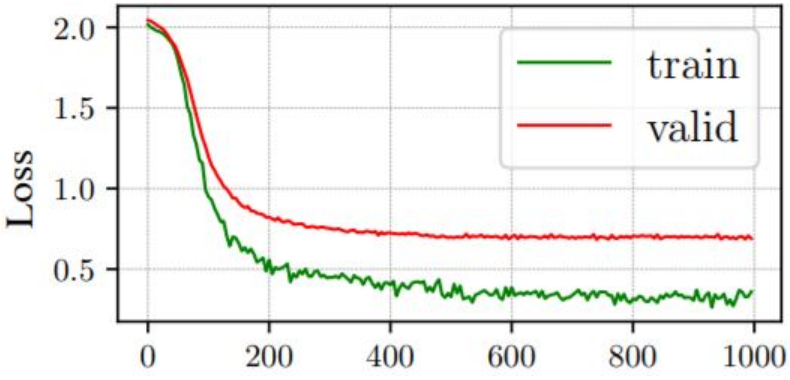
Method	Cora	Citeseer	Pubmed
GCN [20]	81.5	70.3	79.0
GAT [35]	83.0±0.7	72.5±0.7	79.0±0.3
APPNP [21]	83.8±0.3	71.6± 0.5	79.7 ± 0.3
Graph U-Net [12]	84.4±0.6	73.2±0.5	79.6±0.2
SGC [39]	81.0 ±0.0	71.9 ± 0.1	78.9 ± 0.0
MixHop [1]	81.9± 0.4	71.4±0.8	80.8±0.6
GMNN [31]	83.7	72.9	81.8
GraphNAS [13]	84.2±1.0	73.1±0.9	79.6±0.4
GraphSAGE [17]	78.9±0.8	67.4±0.7	77.8±0.6
FastGCN [7]	81.4±0.5	68.8±0.9	77.6±0.5
VBAT [9]	83.6±0.5	74.0±0.6	79.9±0.4
G^3NN [25]	82.5±0.2	74.4±0.3	77.9 ±0.4
GraphMix [36]	83.9±0.6	74.5±0.6	81.0±0.6
DropEdge [32]	82.8	72.3	79.6
GRAND_dropout	84.9±0.4	75.0±0.3	81.7±1.0
GRAND_DropEdge	84.5±0.3	74.4±0.4	80.9±0.9
GRAND_GCN	84.5±0.3	74.2±0.3	80.0±0.3
GRAND_GAT	84.3±0.4	73.2± 0.4	79.2±0.6
GRAND	**85.4±0.4**	**75.4±0.4**	**82.7±0.6**
w/o CR	84.4±0.5	73.1±0.6	80.9±0.8
w/o mDN	84.7±0.4	74.8±0.4	81.0±1.1
w/o sharpening	84.6±0.4	72.2±0.6	81.6±0.8
w/o CR & DN	83.2±0.5	70.3±0.6	78.5±1.4

Table 1: Overall classification accuracy (%).

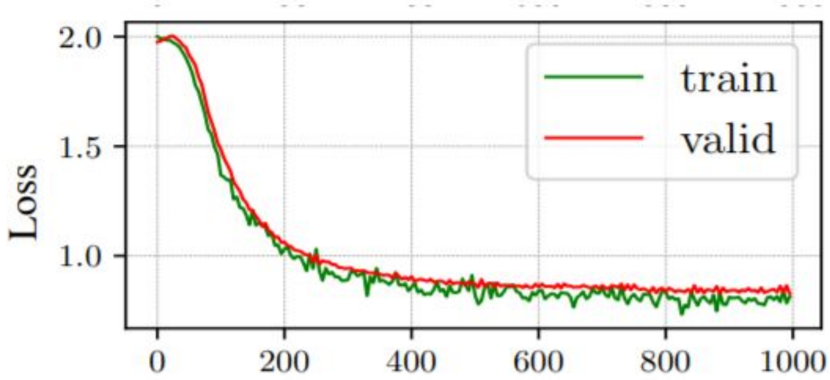
Generalization Analysis



(b) Without CR

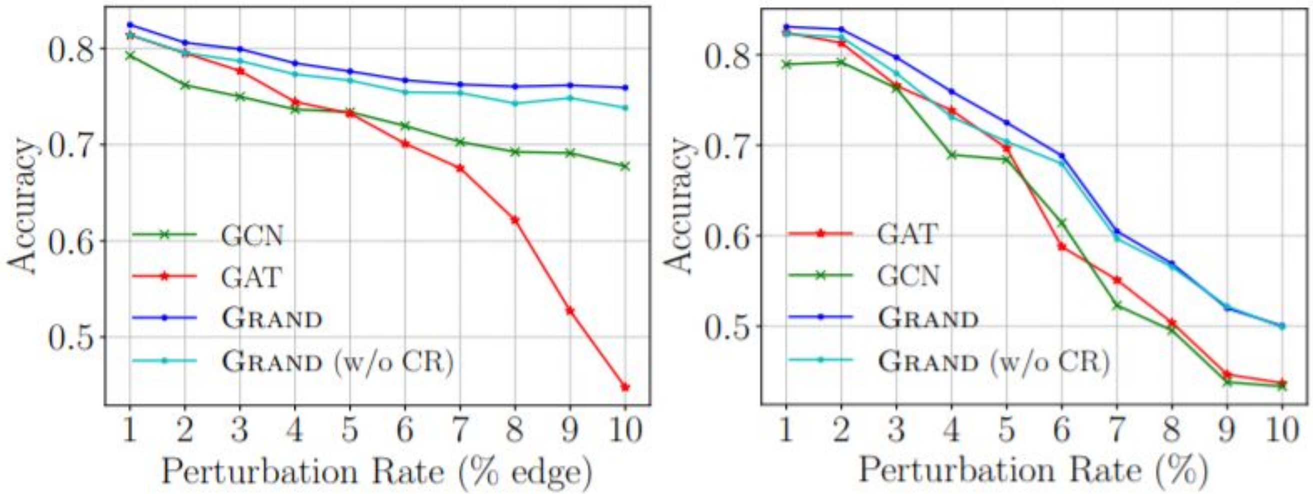


(a) Without RP



(c) GRAND(with RP and CR)

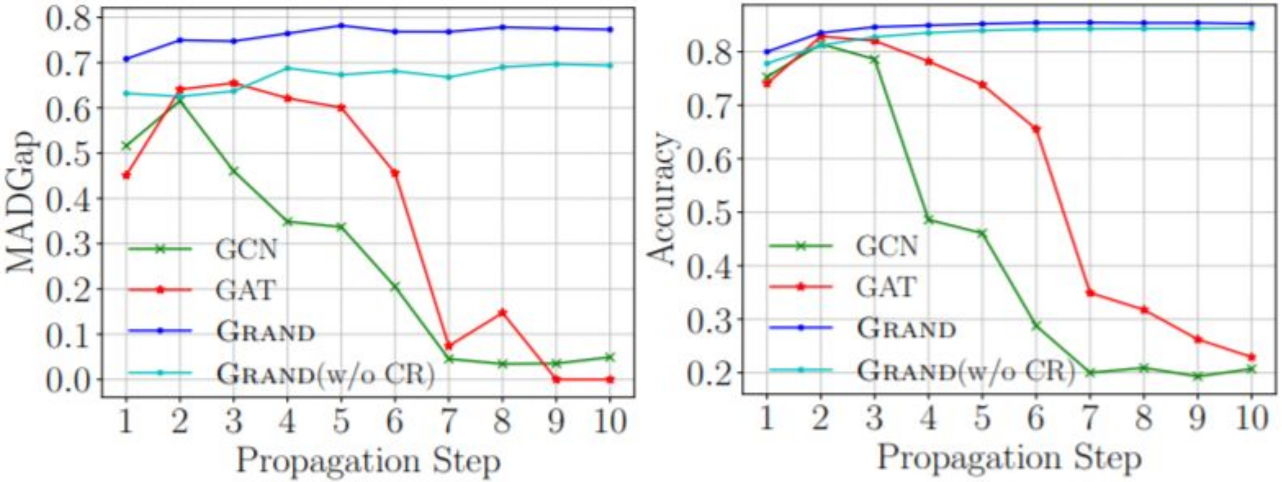
Robustness Analysis



(a) Random Attack

(b) Metattack

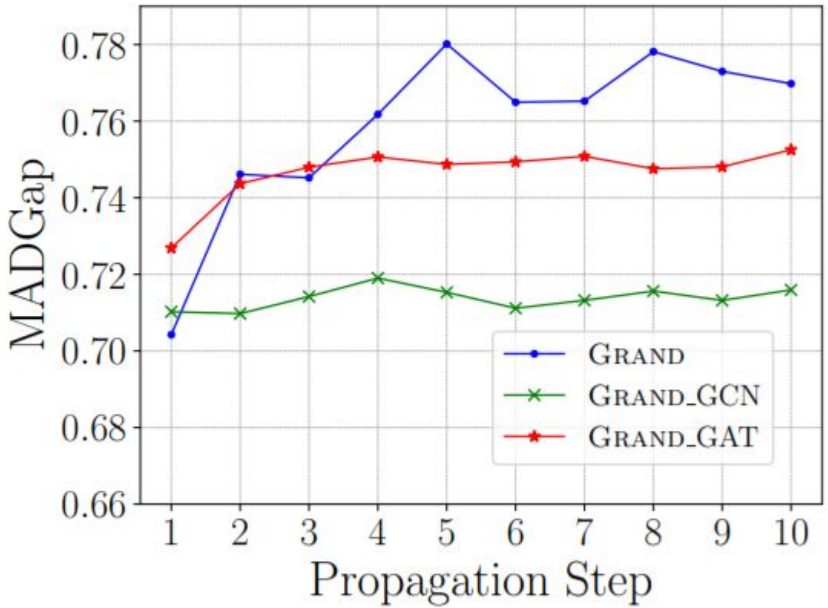
Over-smoothing analysis



(a) MADGap (b) Classification Result

Other results presented in the paper

Over-smoothness of GRAND and its variants(on Cora)



Other results presented in the paper

Classification Accuracy of GRAND on large datasets

Method	Cora Full	Coauthor CS	Coauthor Physics	Amazon Computer	Amazon Photo	Aminer CS
GCN	62.2 ± 0.6	91.1 ± 0.5	92.8 ± 1.0	82.6 ± 2.4	91.2 ± 1.2	49.9 ± 2.0
GAT	51.9 ± 1.5	90.5 ± 0.6	92.5 ± 0.9	78.0 ± 19.0	85.7 ± 20.3	49.6 ± 1.7
GRAND	**63.5 ±0.6**	**92.9 ± 0.5**	**94.6 ± 0.5**	**85.7 ± 1.8**	**92.5 ± 1.7**	**52.8 ± 1.2**

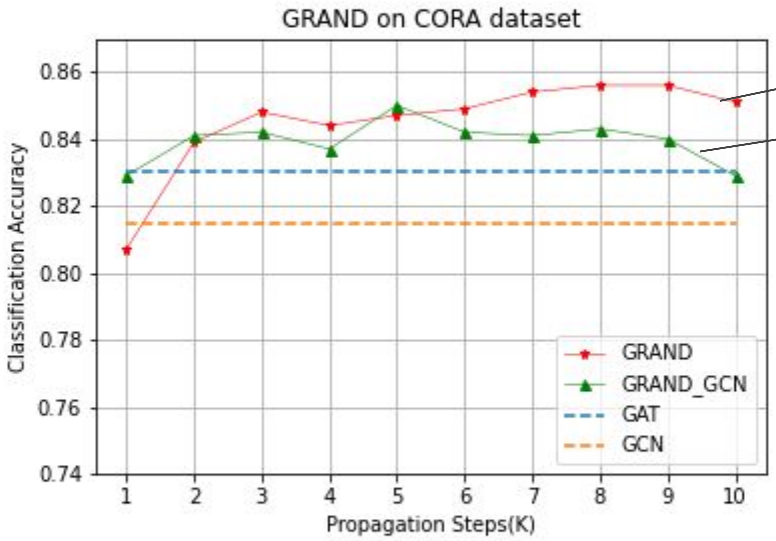
Experiments

The following experiments were conducted:

- MLP v/s GCN as classification network
- Classification accuracy v/s {K, S}
- Sensitivity wrt CR loss coefficient λ

1. MLP v/s GCN

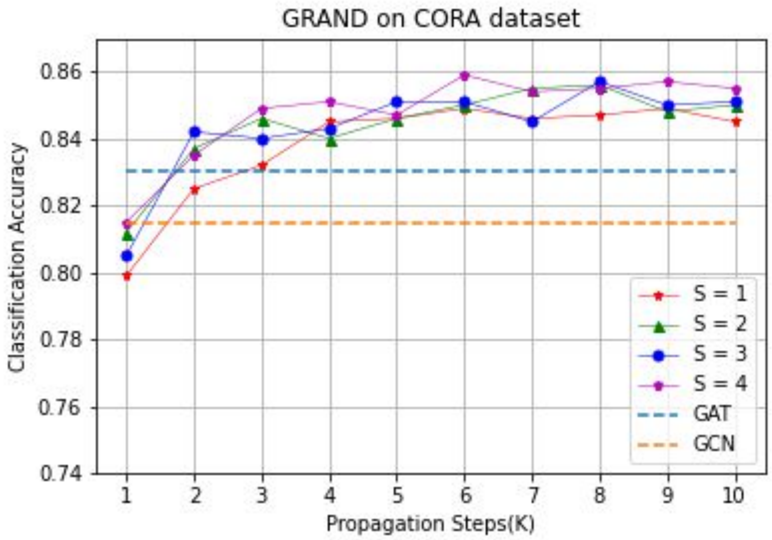
Effect of using an MLP vs GCN as the classification network



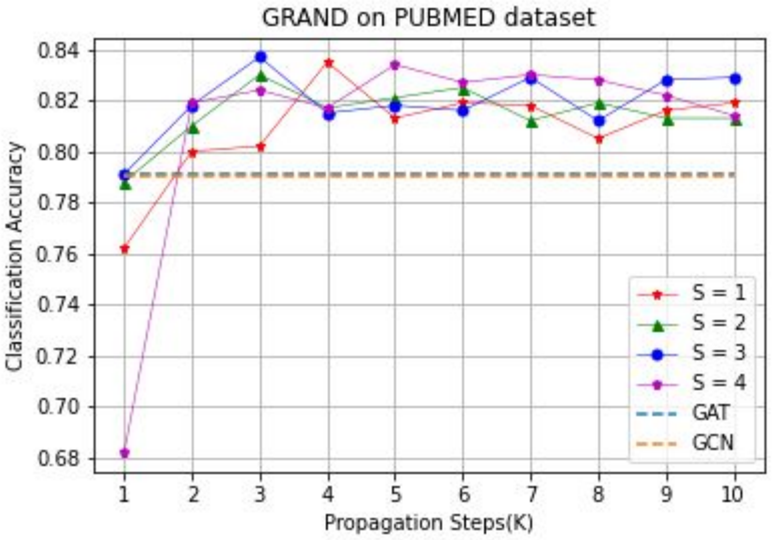
GRAND very clearly outperforms **GRAND_GCN** in terms of *classification accuracy*

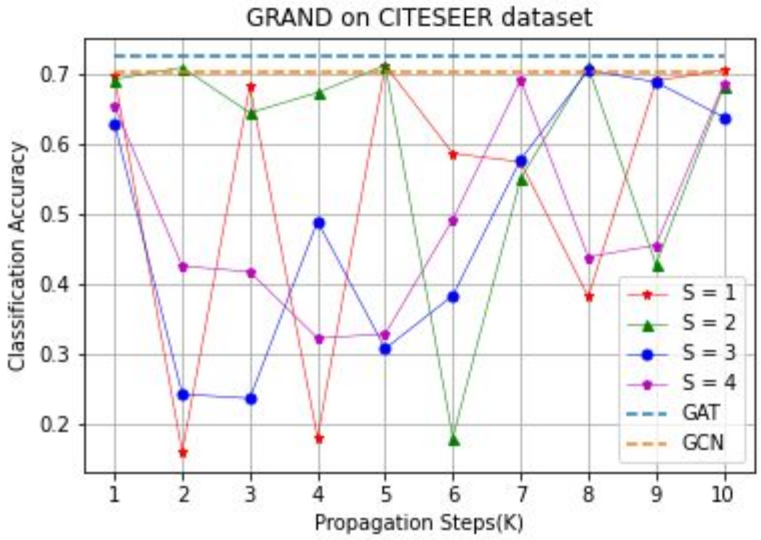
2. Classification Accuracy v/s {K, S}

(i) Effect of K(propagation order) and S(number of data augmentations) on Classification Accuracy on GRAND(DropNode data augmentation)



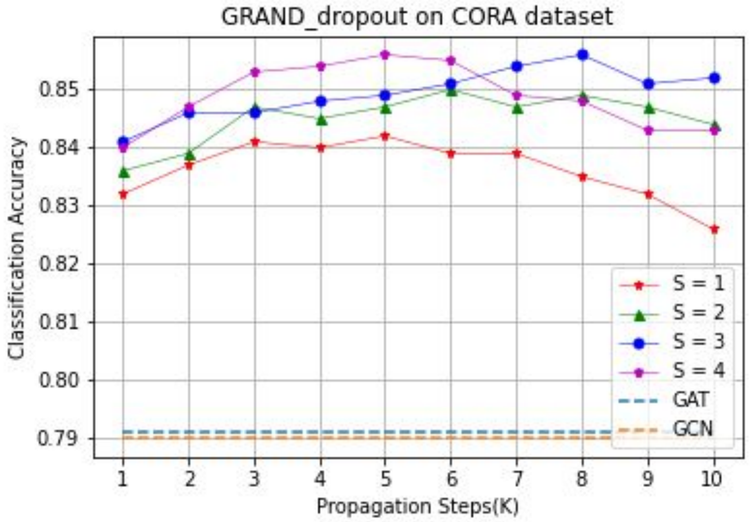
2. Classification Accuracy v/s {K, S}

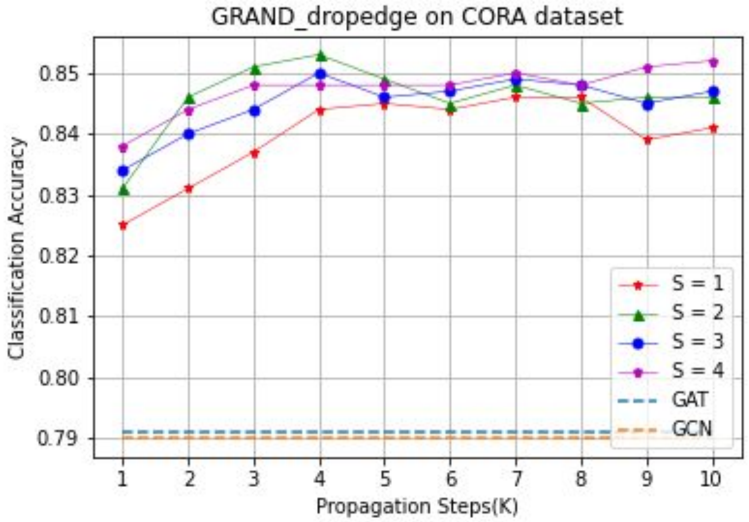




2. Classification Accuracy v/s {K, S}

(ii) Effect of K(propagation order) and S(number of data augmentations) on Classification Accuracy on GRAND_dropout and GRAND_dropedge(alternative data augmentation techniques)

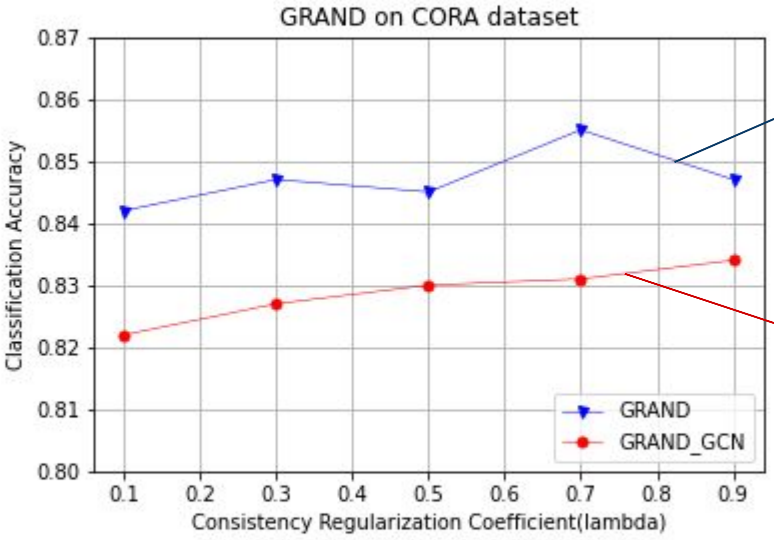




3. Sensitivity wrt ƛ

Classification Accuracy v/s ƛ

Consistency Regularization Loss Coefficient



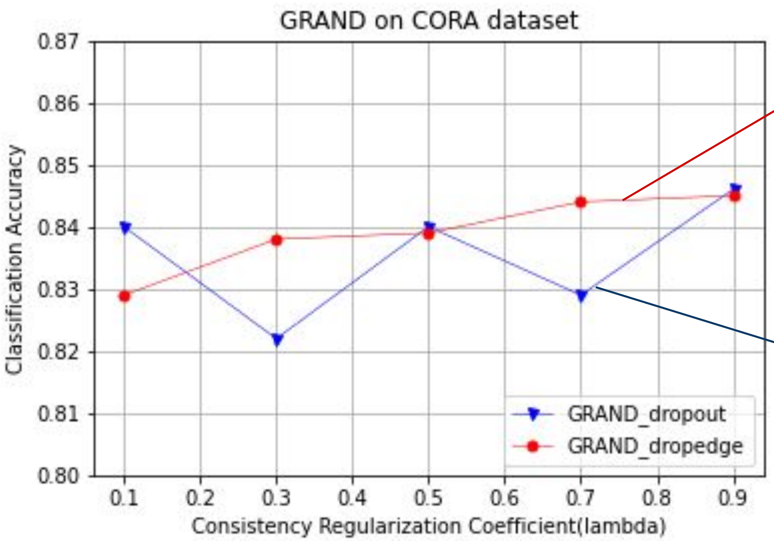
MLP classification network

Both using DropNode for data augmentation

GCN classification network

3. Sensitivity wrt λ

Classification Accuracy v/s λ

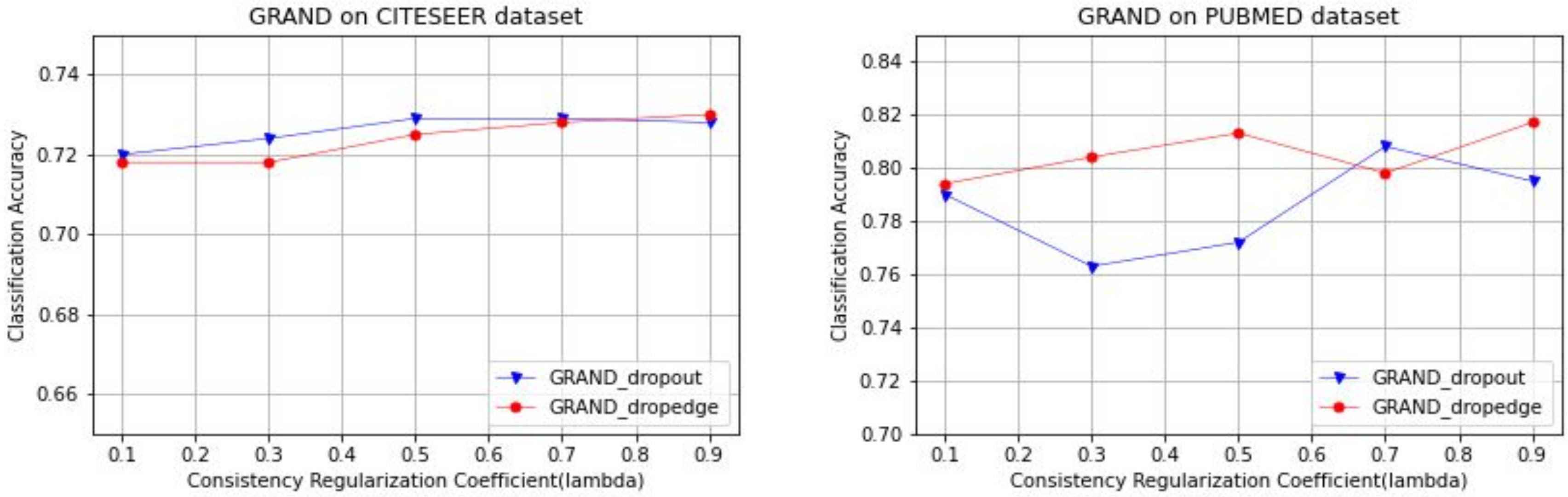


DropEdge data augmentation

Both using MLPs as classification networks

Dropout data augmentation

3. Sensitivity wrt λ



Ablation Study

The effect of the absence of the following parameters was studied:

- w/o consistency regularization(CR)
- w/o multiple dropnode(mDN)
- w/o sharpening
- w/o consistency regularization(CR) and dropnode(DN)

Ablation Study

these are classification accuracies

Method	Cora	Pubmed	Citeseer
w/o CR (λ=0)	0.841	0.811	0.728
w/o mDN (S=1)	0.85	0.80	0.744
w/o sharpening (T=1)	0.844	0.816	0.578
w/o CR & DN (λ=0, δ=0)	0.835	0.787	0.597

END

Thank You!