# Graph Random Neural Network for Semi-Supervised Learning on Graphs

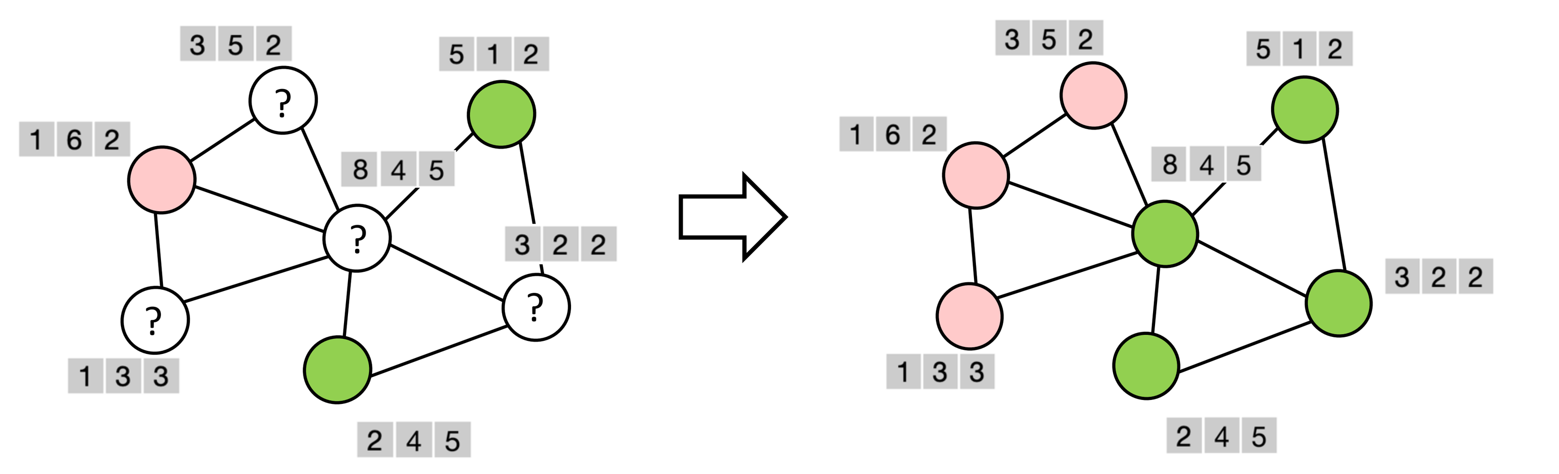

Tsinghua University

Wenzheng Feng, Jie Zhang, Yuxiao Dong, Yu Han, Huanbo Luan, Qian Xu, Qiang Yang, Evgeny Kharlamov, Jie Tang

fwz17@tstinghua.edu.cn, {zhangjie.exe, ericdongyx, yhanthu, luanhuanbo}@gmail.com, {qianxu, qiangyang}@webank.com, evgeny.Kharlamov@de.bosch.com, jietang@tsinghua.edu.cn


NEURAL INFORMATION PROCESSING SYSTEMS

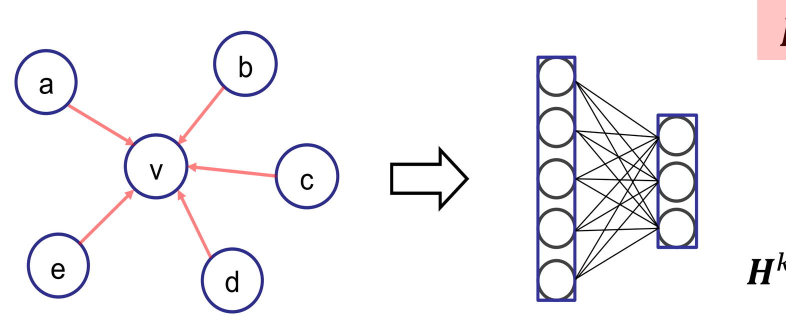
## Semi-Supervised Learning on Graphs




**Input:** a partially labeled & attributed graph

**Output:** infer the labels of unlabeled nodes

## Representative Method: Graph Convolution Network




node $v$'s embedding at $k+1$ — non-linear activation function (e.g. ReLU)

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

normalized Laplacian matrix

$$H^{k+1} = \sigma\left(W^k \sum_{u \in N(v) \cup v} \frac{H_u^k}{\sqrt{|N(u)||N(v)|}}\right)$$

the neighbors of node $v$

- Each node is highly dependent with its neighborhoods, making GNNs **non-robust** to noises.
- Stacking many GNNs layers may cause **over-smoothing.**
- Under semi-supervised setting, standard training method is easy to **over-fit** the scarce label information.

## Summary


- We propose Graph Random Neural Network (GRAND), a simple yet effective framework for semi-supervised learning on graphs.
- GRAND adopts a simple Random Propagation strategy to augment each node stochastically, wherein each node's features are randomly dropped either partially or entirely, after which the perturbed feature matrix is propagated over the graph.
- To improve model's generalization capacity, GRAND utilizes consistency regularization strategy to optimize the prediction consistency among multiple augmentations produced by Random Propagation.
- We theoretically analyze the regularization effects of the proposed random propagation and consistency regularization strategy.
- We empirically show that GRAND mitigates the issue of over-smoothing and non-robustness, exhibiting better generalization than existing GNNs.
- GRAND outperforms **14** GNN baselines on three graph benchmark datasets.


## Graph Random Neural Network (GRAND)

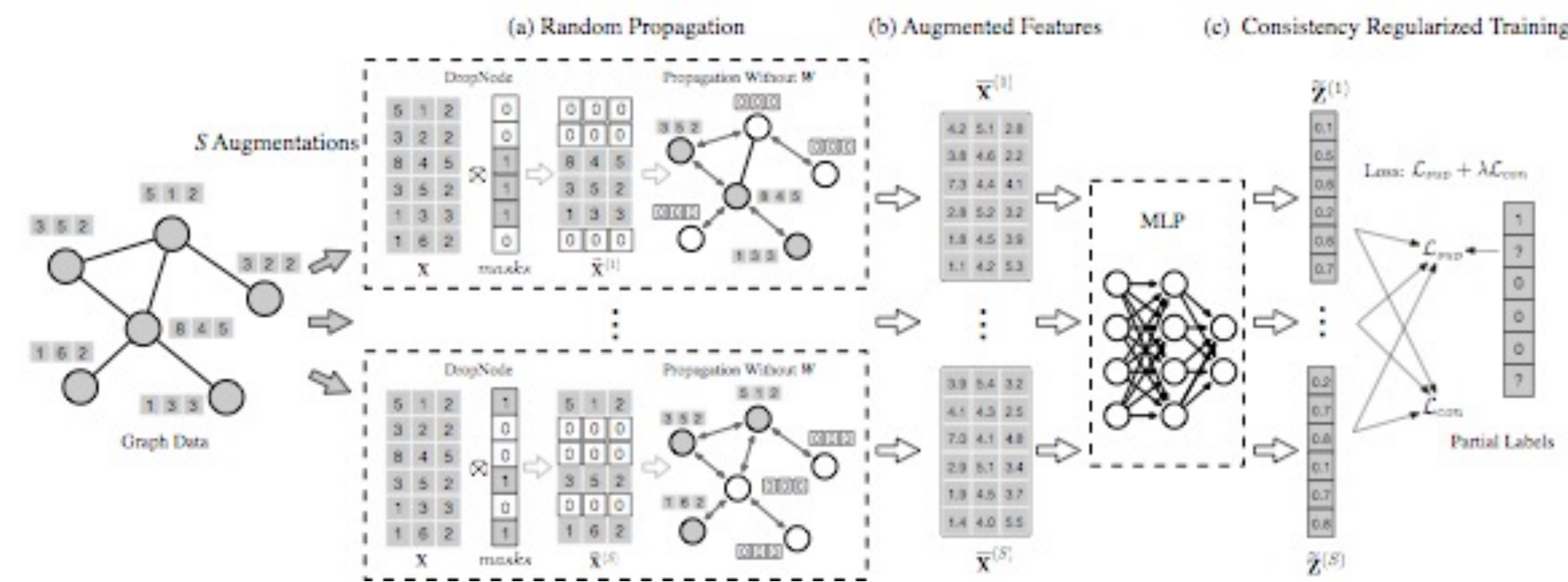



Illustration of GRAND with DropNode as the perturbation method. GRAND designs random propagation (a) to generate multiple graph data augmentations (b), which are further used as consistency regularization (c) for semi-supervised learning.

## Training Algorithm of GRAND

**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 $\delta$, learning rate $\eta$, an MLP model: $f_{mlp}(\mathbf{X}, \Theta)$.

**Output:**
Prediction $\mathbf{Z}$.

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

## Theoretical Analysis

**Theorem 1.** *In expectation, optimizing the unsupervised consistency loss $\mathcal{L}_{con}$ is approximate to optimize a regularization term:* $\mathbb{E}_\epsilon(\mathcal{L}_{con}) \approx \mathcal{R}^c(\mathbf{W}) = \sum_{i=0}^{n-1} z_i^2 (1 - z_i)^2 \text{Var}_\epsilon \left(\overline{\mathbf{A}}_i \widetilde{\mathbf{X}} \cdot \mathbf{W}\right)$.

**Theorem 2.** *In expectation, optimizing the perturbed classification loss $\mathcal{L}_{sup}$ is equivalent to optimize the original loss $\mathcal{L}_{org}$ with an extra regularization term $\mathcal{R}(\mathbf{W})$, which has a quadratic approximation form* $\mathcal{R}(\mathbf{W}) \approx \mathcal{R}^q(\mathbf{W}) = \frac{1}{2} \sum_{i=0}^{m-1} z_i (1 - z_i) \text{Var}_\epsilon \left(\overline{\mathbf{A}}_i \widetilde{\mathbf{X}} \cdot \mathbf{W}\right)$.

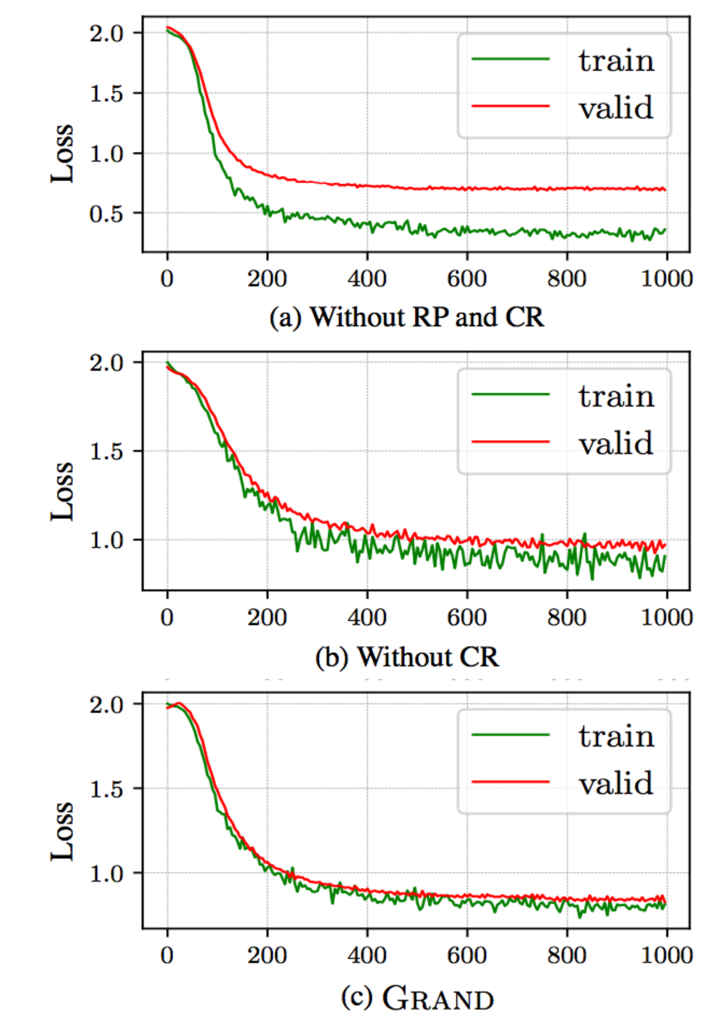
- With Consistency Regularization Loss:
  - Random propagation can enforce the consistency of the classification confidence between each node and its all multi-hop neighborhoods.
- With Supervised Cross-Entropy Loss:
  - Random propagation can enforce the consistency of the classification confidence between each node and its labeled multi-hop neighborhoods.

## Overall Results

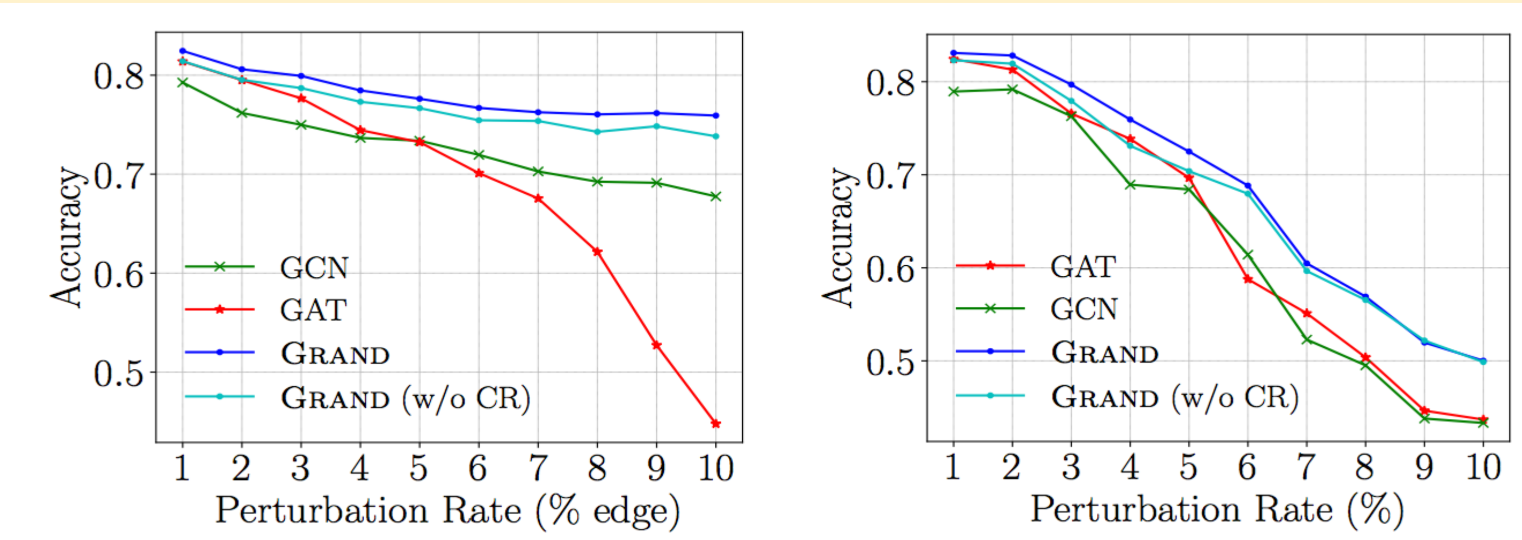
| Method | Cora | Citeseer | Pubmed |
|---|---|---|---|
| GCN [24] | 81.5 | 70.3 | 79.0 |
| GAT [41] | 83.0±0.7 | 72.5±0.7 | 79.0±0.3 |
| APPNP [25] | 83.8±0.3 | 71.6± 0.5 | 79.7 ± 0.3 |
| Graph U-Net [13] | 84.4±0.6 | 73.2±0.5 | 79.6±0.2 |
| SGC [45] | 81.0 ±0.0 | 71.9 ± 0.1 | 78.9 ± 0.0 |
| MixHop [1] | 81.9± 0.4 | 71.4±0.8 | 80.8±0.6 |
| GMNN [36] | 83.7 | 72.9 | 81.8 |
| GraphNAS [14] | 84.2±1.0 | 73.1±0.9 | 79.6±0.4 |
| GraphSAGE [19] | 78.9±0.8 | 67.4±0.7 | 77.8±0.6 |
| FastGCN [8] | 81.4±0.5 | 68.8±0.9 | 77.6±0.5 |
| VBAT [10] | 83.6±0.5 | 74.0±0.6 | 79.9±0.4 |
| $G^3$NN [29] | 82.5±0.2 | 74.4±0.3 | 77.9 ±0.4 |
| GraphMix [42] | 83.9±0.6 | 74.5±0.6 | 81.0±0.6 |
| DropEdge [37] | 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 (%).

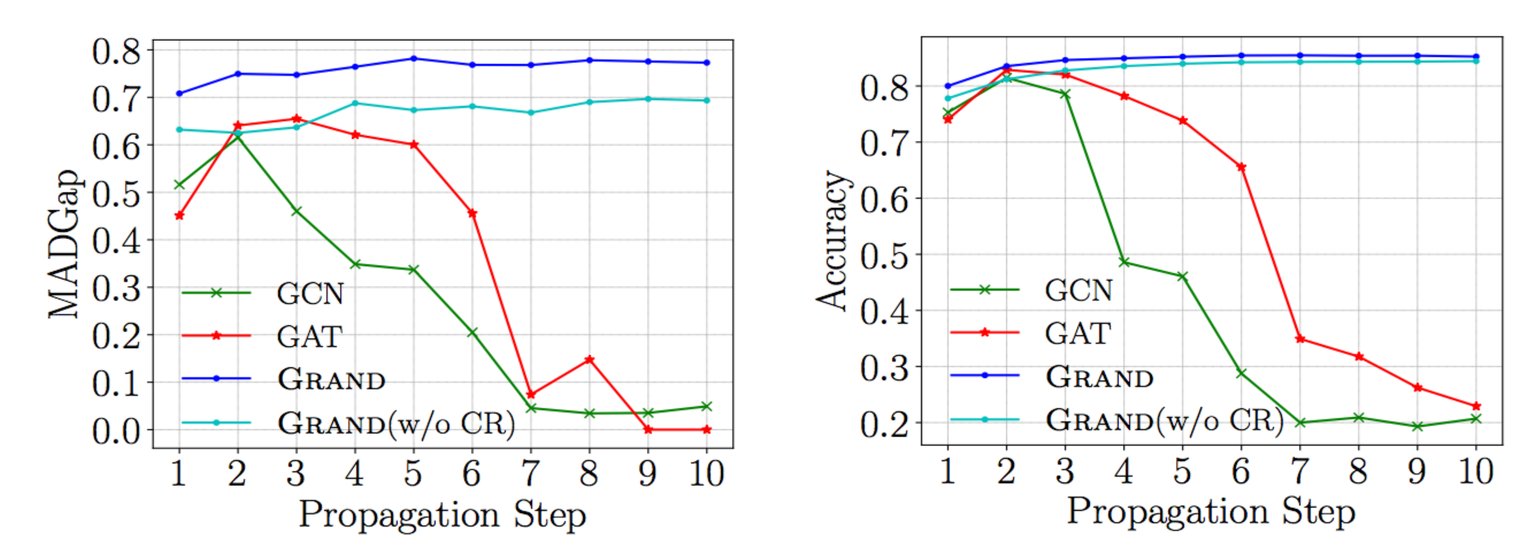
## Generalization Analysis




## Robustness Analysis




## Oversmoothing Analysis




## Results with different label rates

| Dataset | Cora | | | Citeseer | | | Pubmed | | |
|---|---|---|---|---|---|---|---|---|---|
| Label Rate | 1% | 3% | 5% | 1% | 3% | 5% | 0.1% | 0.3% | 0.5% |
| GCN | 62.8±5.3 | 76.1±1.9 | 79.6±2.1 | 63.4±2.9 | 70.6±1.7 | 72.2±1.1 | 71.5±2.1 | 77.5±1.8 | 80.8±1.5 |
| GAT | 64.3±5.8 | 77.2±2.4 | 80.8±2.1 | 64.4±2.9 | 70.4±1.9 | 72.0±1.3 | 72.0±2.1 | 77.6±1.6 | 80.6±1.2 |
| GRAND | **69.1±4.0** | **79.5±2.2** | **83.0±1.6** | **65.3±3.3** | **72.3±1.8** | **73.8±0.9** | **74.7±3.4** | **81.4±2.1** | **83.8±1.3** |


Tsinghua University — WeBank 微众银行 — Microsoft — BOSCH Invented for life

arXiv:

Github: