Appendix for Topology-Imbalance Learning for Semi-Supervised Node Classification

Deli Chen[1,2], **Yankai Lin**[1], **Guangxiang Zhao**[2], **Xuancheng Ren**[2],
Peng Li[1], **Jie Zhou**[1], **Xu Sun**[2]
[1]Pattern Recognition Center, WeChat AI, Tencent Inc., China
[2]MOE Key Lab of Computational Linguistics, School of EECS, Peking University
{delichen, yankailin, patrickpli,withtomzhou}@tencent.com
{zhaoguangxiang,renxc,xusun}@pku.edu.cn

A More Details of Dataset and Experiments

Linux Server We run all the experiments on a Linux server, some important information is listed:

- CPU: Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz $\times$ 40
- GPU: NVIDIA GeForce RTX2080TI-11GB $\times$ 8
- RAM: 125GB
- cuda: 11.1

Python Package We implement all deep learning methods based on Python3.7. The experiment code is included in the supplementary materials. The versions of some important packages are listed:

- torch [11]: 1.9.1+cu111
- torch-geometric [4]: 2.0.1
- torch-cluster:1.5.9
- torch-sparse: 0.6.12
- scikit-learn: 1,0
- numpy:1.20.3
- scipy:1.7.1

Datasets The statistical information of our experimental datasets are shown in Table 1. All these data are publicly available and the URLs listed as follows:

- Planetoid Citation Datasets [13] (*CORA/CiteSeer/PubMed*): https://github.com/rusty1s/pytorch_geometric/blob/master/torch_geometric/datasets/planetoid.py
- Amazon Co-purchasing Datasets [9] (*Photo/Computers*): https://github.com/rusty1s/pytorch_geometric/blob/master/torch_geometric/datasets/amazon.py
- Reddit Comment Dataset [5]:https://github.com/TUM-DAML/pprgo_pytorch/blob/master/data/get_reddit.md
- MAG-Scholar Dataset [1] (coarse grained version): https://figshare.com/articles/dataset/mag_scholar/12696653/2

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

Table 1: Statistical information about datasets. M indicates million.

Dataset	#Node	#Edge	#Feature	#Class	Training
CORA	2,708	5,429	1,433	7	Transductive
CiteSeer	3,327	4,732	3,703	6	Transductive
PubMed	19,717	44,338	5,00	3	Transductive
Photo	7,487	119,043	745	8	Transductive
Computers	13,381	245,778	767	10	Transductive
Reddit	232,965	11,606,919	602	41	Inductive
MAG-Scholar	10.5145M	132.8176M	2.7842M	8	Inductive

Table 2: Dataset Topology-Imbalance Level

$\sum_{v\in\mathcal{L}} \boldsymbol{T}_v$	LOW	MIDDLE	HIGH
CORA	$4.26_{\pm 0.27}$	$6.03_{\pm 0.21}$	$7.39_{\pm 0.43}$
CiteSeer	$1.19_{\pm 0.11}$	$2.26_{\pm 0.01}$	$4.37_{\pm 0.23}$
Pubmed	$0.14_{\pm 0.02}$	$0.25_{\pm 0.01}$	$0.42_{\pm 0.05}$

Dataset Splitting In training, we run 5 different random splittings for each dataset to relieve the randomness introduce by the training set selection following Shchur et al. [14]. We repeat experiments 3 times for each splitting to relieve the training splitting. The final performance (weighted F1, macro F1, and the standard deviation) is calculated based on the 15 repeated experiments. The dataset splitting seed list is $[0, 1, 2, 3, 4]$; the model training random seed list is: $[0, 1, 2]$.

Method Hyperparameters For all encoders ($\mathcal{F}$ and $\mathcal{F}'$), we stacked two GNN or linear layers with the ReLU [10] activation function[1]. All the hyper-parameters are tuned on the validation set. The tuning range of dataset-specific hyperparameters is as follows:

- PageRank teleport probability α: $[0.05, 0.1, 0.15, 0.2]$;
- Dimension of hidden layer: $[16, 32, 64, 128, 256]$;
- Lower bound of the cosine annealing w_{min}: $[0.25, 0.5, 0.75]$;
- Upper bound of the cosine annealing w_{max}: $[1.25, 1.5, 1.75]$;

Training Setting We take the Adam [7] as the model optimizer. The learning ratio begins to decay after 20 epochs with a ratio of 0.95. We early stop the training process if there is no improvement in 20 epochs. The tuning range of dataset-specific hyperparameters is as follows:

- Learning Rate:$[0.005, 0.0075, 0.01, 0.015]$,
- Dropout Probability: $[0.2, 0.3, 0.4, 0.5, 0.6]$.

B Supplement to the ReNode Method

Apart from the relative ranking re-weight method in ReNode, we also tried to adjust the training weight based on the following scheduling methods:

- Linear decay based on the original node Totoro values;
- Linear decay based on the rank of node Totoro values;
- Discrete values for different nodes with a piece-wise function;

Among all these methods, the presented cosine annealing method works best. We analyze the reason lies in that, PageRank is proposed for node ranking; hence adjusting weights based on the original

[1]Except the SGC model, which increases the power iteration times of the normalized adjacency matrix to replace stacking GNN layers.

values is not robust and can be largely affected by outliers. Comparing to the linear decay schedule, the cosine schedule methods pay more attention to nodes with middle-level conflict, distinguishing which is of great importance for the model training.

The ReNode method assigns more weights to nodes far away from the graph class boundaries, which it is different from methods used in metric learning [15, 3] or contrastive learning [6, 12] that pay more attention to the 'hard' samples closing to class boundaries. In semi-supervised node classification, most message-passing based GNN model (e.g. GCN) relies on smoothing the adjacent nodes to transfer the category information from the labeled nodes to the unlabeled nodes [8, 2]. Thus, the 'easy' labeled nodes far away from the class boundaries are expected to better represent class prototypes. Enlarging the training weights of those 'hard' nodes that are close to the class boundaries makes it easier to confuse the class prototype with others. Besides, the labeling size in semi-supervised learning is much smaller than supervised learning (usually 20 nodes per class) and usually, the training nodes are sampled randomly. Hence, a very likely scenario is that the 'hard' samples for some categories are very close to the true class boundaries, while the 'hard' samples for other categories are far away from the true class boundaries. Relying on these 'hard' nodes to decide decision boundaries will cause a large shift of decision boundaries from the true ones.

C Settings of Dataset Topology-Imbalance Levels

In Section 3, we evaluate the model performance under different levels of topology imbalance. We introduce the settings for the topology-imbalance levels. For each experiment dataset, we randomly sampled 100 training sets, and calculate the dataset overall conflict as introduced in Section 2.3. Then we choose the 3 training sets with the highest/middle/lowest overall conflict as the high/middle/low-level topology-imbalance setting and report the average results on the 3 training sets for each dataset. The specific conflict values of different levels are displayed in Table 2.

D Submission Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The main claims stated in the abstract and introduction accurately reflect the paper's contributions and scope.
   (b) Did you describe the limitations of your work? [Yes] We discuss the limitation in Section 4.2.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] We think that our proposal has no obvious potential negative societal effect.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We have read ethics review guidelines and ensure that our paper conforms to them.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We include them in the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We describe them in detail in the paper main body and the appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report error bars and the random seed for all experiments in the paper main body and the appendix.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We include them in the appendix.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite all the existing assets used in this work.
   (b) Did you mention the license of the assets? [Yes] We mention the license in appendix.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A] We have no new assets.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] All the datasets we use is open-source and can be obtained from their public release.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] the datasets we use has no personally identifiable information or offensive content.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

References

[1] Aleksandar Bojchevski, Johannes Klicpera, Bryan Perozzi, Amol Kapoor, Martin Blais, Benedek Rózemberczki, Michal Lukasik, and Stephan Günnemann. Scaling Graph Neural Networks with Approximate PageRank. In *the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2020*, pages 2464–2473. ACM, 2020.

[2] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and Relieving the Over-smoothing Problem for Graph Neural Networks from the Topological View. In *the 34th AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 3438–3445. AAAI Press, 2020.

[3] Yueqi Duan, Wenzhao Zheng, Xudong Lin, Jiwen Lu, and Jie Zhou. Deep Adversarial Metric Learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 2780–2789. Computer Vision Foundation / IEEE Computer Society, 2018.

[4] Matthias Fey and Jan E. Lenssen. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[5] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1024–1034, 2017.

[6] Yannis Kalantidis, Mert Bülent Sariyildiz, Noé Pion, Philippe Weinzaepfel, and Diane Larlus. Hard Negative Mixing for Contrastive Learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[7] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[8] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper Insights into Graph Convolutional Networks for Semi-supervised Learning. In *the 32nd AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3538–3545. AAAI Press, 2018.

[9] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based Recommendations on Styles and Substitutes. In *the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, pages 43–52. ACM, 2015.

[10] Vinod Nair and Geoffrey E Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 807–814. Omnipress, 2010.

[11] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic Differentiation in PyTorch. In *NeuriIPS 2017 Workshop on Autodiff*, 2017.

[12] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive Learning with Hard Negative Samples. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[13] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective Classification in Network Data. *AI magazine*, 29(3):93–93, 2008.

[14] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of Graph Neural Network Evaluation. *arXiv preprint: 1811.05868*, 2018.

[15] Wenzhao Zheng, Zhaodong Chen, Jiwen Lu, and Jie Zhou. Hardness-Aware Deep Metric Learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 72–81. Computer Vision Foundation / IEEE, 2019.