Decoupled Graph Neural Networks based on Label Agreement Message Propagation

Zhicheng An & Yue Wang & Shao-Lun Huang [*]
Tsinghua University
azc19@mails.tsinghua.edu.cn, wangyue@mail.tsinghua.edu.cn, twn2gold@gmail.com

Zhengwei Wu & Binbin Hu & Zhiqiang Zhang & Jun Zhou
Ant Group
{zejun.wzw, bin.hbb, lingyao.zzq, jun.zhoujun}@antfin.com

Abstract

Decoupling has become a new paradigm in Graph Neural Networks (GNNs) for its effectiveness and scalability. However, this paradigm still faces two several restrictions: unsatisfying propagation, caused by noisy or confused edges, could greatly degrade model performance; fixed aggregation schema with the same propagation steps and the same combination weights for each node limit achieving optimal performance. To address these problems, we propose a novel decoupled graph model named LA-DGNN based on label agreement message propagation and combine the intermediate feature after each propagation step as input. In our method, we decouple the graph model which trains a base predictor based on multi-layer perceptrons with a pre-step to propagate features and a post-step to propagate labels. We utilize an auxiliary label agreement model to generate proper edge weights to promote reliable propagation. When training the base predictor, we concatenate all intermediate features after each propagation step to make the model dynamically learn information of neighbors at different distances. Extensive experiments on five real-world datasets demonstrate that our method achieves superior performance over all baseline methods in terms of node classification accuracy.

1 Introduction

Graph Neural Networks (GNNs) have achieved great success in a wide range of graph-based applications, such as node classification, graph classification, link prediction and community detection (Li et al., 2018; Knyazev et al., 2019; He et al., 2020; Bakshi et al., 2018). Most GNNs follow the paradigm that features are transformed and aggregated via graph convolution layers to generate node representations. Through K graph convolution layers, nodes obtain information from their K-hop neighborhoods which are called Receptive Field. However, these models face the challenge of receptive field restriction. The number of GNN layers grows with the size of receptive field, resulting in high computation and memory cost. What's worse, GNNs occur notorious over-smoothing issue while directly applying multiple layers (Li et al., 2019).

Many recent advancing works try to decouple the feature transformation and neighborhood aggregation in each convolution layer to access to more scalable and efficient models. For example, SGC (Wu et al., 2019) successively removes nonlinearities and collapsing weight matrices between consecutive layers to reduce excess complexity. APPNP (Klicpera et al., 2019) separates the feature propagation and neural network training to achieve a much larger receptive field size. They derive an improved propagation schema based on personalized PageRank to permit the use of far more propagation steps without leading to

[*] Corresponding author

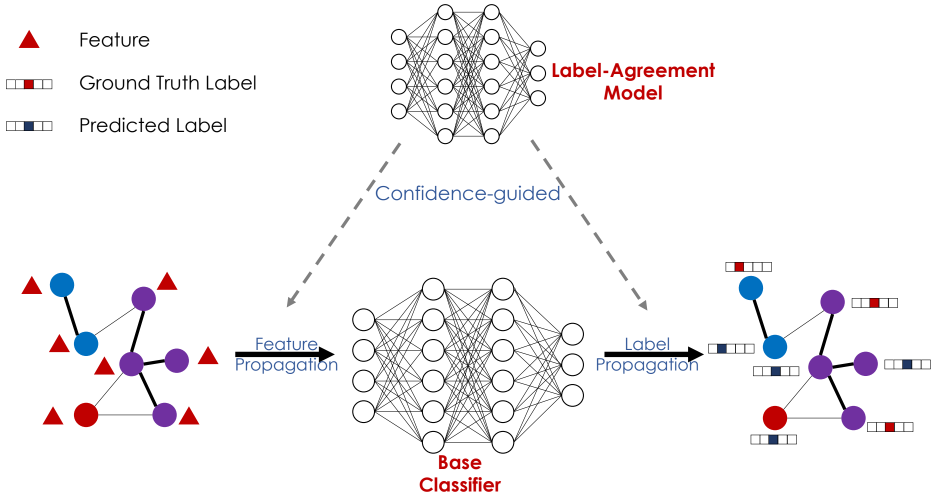

Figure 1: Illustration of LA-DGNN

over-smoothing. On the contrary, C&S (Huang et al., 2021) propagates label messages. They train a base predictor with node features that ignores graph structure and apply two post operation based on label propagation (Zhou et al., 2003; Wang & Zhang, 2008) which propagate residual errors and labels separately.

Both feature propagation and label propagation are based on the homophily hypothesis that features and labels vary smoothly over the edges of the graph. However, this hypothesis is not fully matched in practice for graph edges are from noisy sources. Therefore, some edges' corresponding labels are not the same. Besides, these models treat each propagation equally that use a hyper-parameter to mix current information and neighborhood information. However, neighbors at different distances have different importance for nodes. To address these two problems, based on previous advancements, we propose a novel decoupled graph model named LA-DGNN. In our model, we first train a label agreement model based on the labels in training set to generate proper edge weight for each edge. Then we propagate features according to the weighted adjacent matrix and concatenates intermediate features for each node after every propagation step. The aggregated features serve as input for a base predictor based on multi-layer perceptrons. After training, a post label propagation is applied to smooth predictions and leverage known labels.

2 Methodology

We start with notations used through this paper. Let $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ denotes an undirected graph with nodes $\mathcal{V}$ and edges $\mathcal{E}$. The nodes in G are described by the feature matrix $\mathbf{X} \in \mathbb{R}^{n \times h}$, where n is the number of nodes and h is the dimension of features. Only a subset of nodes $\mathcal{V}^L$ have observed labels and the goal is to infer the labels of the remaining nodes $\mathcal{V}^U$.

Our model follows the idea of decoupled GNNs, which treats message propagation as pre- and post- operations. The illustration of LA-DGNN is demonstrated in Figure 1. We train a classifier based on the augmented features after a feature propagation process. Then, we apply a label propagation process to smooth the prediction. In order to ensure the correct propagation paths, we train a label agreement prediction model based on the known labels. Aiming at getting the best receptive field for each node, we combine the features after every propagation step into a vector to feed into the base classifier.

2.1 Label agreement model

Most existing graph learning methods rely on the homophily hypothesis to incorporate graph structure into model design. Therefore, the noisy edges which connect two nodes with different labels will damage the performance. Take LP (Zhou et al., 2003) as an example, its performance on graphs without noisy edges achieves 17.1%, 8.9%, 18.7% improvement compared with graphs with noisy edges on Cora, Citeseer and Pubmed datasets under standard division. A series of methods try to reduce the impact of the noisy edges. For instance, GAT (Velickovic et al., 2018) utilizes the features of two connected nodes to learn the edge weights for convolution operation.

We borrow the idea from GAM (Stretcu et al., 2019) and utilize the edge weight learning process. The features of two neighboring nodes are used as input to predict the label consistency of the two nodes. First, we embed the node features into a hidden space based on multi-layer perceptrons. $\mathbf{e}_i = mlp(\mathbf{x}_i)$, where $\mathbf{x}_i$ represents the feature vector of node i. Then, we apply subtraction and square operations between the two embeddings to eliminate the influence of two nodes' order, $\mathbf{e}_{ij} = (\mathbf{e}_i - \mathbf{e}_j)^2$. The calculation of the aggregated embeddings $\mathbf{e}_{ij}$ can be considered to measure the distance between node i and node j in the hidden space, which is an intuitive representation of the agreement probability (Stretcu et al., 2019). Finally, the aggregated embeddings are fed into a linear classifier to predict the confidence that the two nodes have the same label, $s = cls(\mathbf{e}_{i,j})$. We use binary cross entropy loss with logits ℓ_{bce} as the loss function of the agreement model defined as:

$$\mathcal{L}_{\mathrm{LA}} = \sum_{i \in \mathcal{V}^L, j \in \mathcal{V}^L, ij \in \mathcal{E}} \ell_{bce}(cls(mlp(\mathbf{e}_i, \mathbf{e}_j)), \mathbb{I}_{y_i = y_j}) \tag{1}$$

2.2 Confidence-guided feature propagation

The output confidence scores of the label agreement model reflects the similarity of two connected nodes, which are ideal edge weights on graphs. Appropriate edge weights will promote closer combination of similar nodes and weaken the effect of noisy edges. We use the output confidence to build the confidence-guided adjacency matrix $\tilde{\mathbf{A}}$ in which each non-zero elements $\mathbf{A}_{ij}$ is the sigmoid score predicted by the label agreement model using the feature of node i and node j, $\tilde{\mathbf{A}} = Sigmoid(LA - Model(\mathbf{X}, \mathcal{E}))$. As the previous convention, we apply the symmetric normalization adjacency matrix $\hat{\mathbf{A}}$ from $\tilde{\mathbf{A}}$ in message passing process. $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-1/2} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-1/2}$, where $\tilde{\mathbf{D}}^{-1/2}$ denotes the diagonal degree matrix of $\tilde{\mathbf{A}}$. The K-step feature propagation is defined as,

$$\mathbf{X}^{(k)} \leftarrow \hat{\mathbf{A}} \mathbf{X}^{(k-1)}, \forall k = 1, \ldots, K \tag{2}$$

where $\mathbf{X}^{(0)} = \mathbf{X}$.

After K-step propagation, nodes obtain a vector $[\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(K)}]$ that denotes the original feature and features after each propagation step. As we discuss previously, different nodes have different sensitivities to receptive field. In our model, we utilize a simple but effective approach that concatenates the original feature and propagated features as the input. The following classifier is available to features at different steps, which is able to learn the importance of information at different distances automatically and adjust its concentration to the most important part. The augmented node features are produced as:

$$\hat{\mathbf{X}} = [\mathbf{X}^{(0)} \| \mathbf{X}^{(1)} \| \ldots \| \mathbf{X}^{(K)}] \tag{3}$$

2.3 Model training

After feature propagation, each node's features are augmented by its neighbors. We use the augmented feature matrix $\hat{\mathbf{X}}$ to train a shallow classifier based on multi-layer perceptrons. The cross entropy measurement between the predicted label distributions $\mathbf{h}_i = mlp(\hat{\mathbf{x}}_i)$ and the ground truth one-hot label vector is adopted as the loss function:

$$\mathcal{L}_{CL} = \frac{1}{L} \sum_{i \in \mathcal{V}^L} \sum_{c=1}^{C} \log \frac{\exp(\mathbf{h}_{ic})}{\sum_{j=1}^{C} \exp(\mathbf{h}_{ij})} \mathbf{y}_{ic}, \tag{4}$$

where L represents the size of training set, C denotes the number of classes and $\hat{\mathbf{x}}_i$ is the i-th row of $\hat{\mathbf{X}}$.

In the model training process, the label agreement model and the base predictor can not only be trained sequentially but also co-trained on small graphs. In the co-trained schema, the total loss is $\mathcal{L} = \mathcal{L}_{\text{CL}} + \beta\mathcal{L}_{\text{LA}}$, where β is the balancing parameter.

2.4 Confidence-guided label propagation

Label smoothing has been found to be an effective way to incorporate label information at inference and improve model performance (Huang et al., 2021). In our method, we also apply a confidence-guided label propagation to further boost the performance. The label score matrix $\mathbf{Z}$ can be obtained by applying a softmax function to the output of the base predictor. In order to directly incorporate labels at inference, the score of observed nodes are replaced with their ground truth as $\mathbf{Z}^L = \mathbf{Y}^L$. We perform a weighted Q-step confidence-guided label propagation based on personalized PageRank to obtain the final predicted label distributions.

$$\mathbf{Z}^{(k+1)} = (1-\alpha)\mathbf{Z}^{(0)} + \alpha\hat{\mathbf{A}}\mathbf{Z}^{(k)} \tag{5}$$

In this equation, $\mathbf{Z}^{(0)} = \mathbf{Z}$ and $\alpha \in (0, 1]$ is the teleport (or restart) probability which allow the nodes to preserve their own information with a certain probability. The classification for node $i \in \mathcal{V}^U$ is $l_j = \arg\max \mathbf{z}_j^{(Q)}$, where $\mathbf{z}_j^{(Q)}$ represents the j-th row of $\mathbf{Z}_j^{(Q)}$.

3 Experiments

We conduct extensive experiments to demonstrate the effectiveness of our methods. The quantitative results and analysis of different evaluations are presented here.

3.1 Datasets and baselines

We adopt five public datasets for evaluation: Cora (Sen et al., 2008), Citeseer (Sen et al., 2008) and Pubmed (Namata et al., 2012) are three classic citation network graphs; Coauthor-CS (Shchur et al., 2018) and Coauthor-Physics (Shchur et al., 2018) are two co-authorship graphs based on the Microsoft Academic Graph. The detailed statistics are summarized in APPENDIX B. As for the train/valid/test splits, we use 60%/20%/20% random splits following (Wang & Leskovec, 2020).

In the experiments, we compare our model with a range of representative methods, including LP (Zhou et al., 2003), GCN (Kipf & Welling, 2017), GAT (Velickovic et al., 2018), JKNet (Xu et al., 2018), APPNP (Klicpera et al., 2019), C&S (Huang et al., 2021), GCN-LPA (Wang & Leskovec, 2020). LP is a parameter-free method. GCN, GAT and APPNP are classical graph models. APPNP propagates features in advance while C&S propagate labels afterwards. GCN-LPA utilizes label propagation output as a regularization.

3.2 Results

We record the accuracy of test set under best performance on valid set. Each experiment is conducted five times and the mean and standard deviation are demonstrated in Table 1.

From the result table, we have several observations. Our proposed model, LA-DGNN outperforms all the baseline methods on each dataset under the semi-supervised setting. For example, our model improves the best-performing baseline model, GCN-LPA, by an absolute 0.84% accuracy on Cora. For the two large coauthor graphs, the accuracy is also improved though slighter compared to the small graphs. Even though LP is a parameter-free method, it still demonstrates competitive performance on most datasets. GCN and GAT obtain worse results for the limited and fixed receptive field. JKNet and APPNP outperform these shallow GNNs for their access to each propagated feature or large receptive fields. C&S and GCN-LPA also achieve higher performance on most datasets especially C&S shows comparable performance to our method, which indicates the key to improving performance is to incorporate label information.

	Cora	Citeseer	Pubmed	*Co*-CS	*Co*-Physics
LP	86.69	71.88	80.50	91.54	95.58
GCN	87.91 ± 0.46	74.89 ± 0.49	86.37 ± 0.27	92.58 ± 0.23	96.14 ± 0.10
GAT	89.20 ± 0.59	75.94 ± 0.94	83.33 ± 0.36	92.44 ± 0.39	96.23 ± 0.15
JKNet	89.13 ± 0.73	76.15 ± 0.35	86.40 ± 0.24	93.64 ± 0.21	96.46 ± 0.13
APPNP	87.84 ± 0.95	76.63 ± 0.72	87.08 ± 0.43	94.28 ± 0.18	96.40 ± 0.15
C&S	87.80 ± 1.03	76.60 ± 0.50	87.43 ± 0.35	94.60 ± 0.23	96.57 ± 0.16
GCN-LPA	88.84 ± 1.07	76.51 ± 0.27	87.11 ± 0.10	93.14 ± 0.19	95.90 ± 0.29
LA-DGNN	$\mathbf{89.68} \pm 0.76$	$\mathbf{77.23} \pm 0.39$	$\mathbf{88.06} \pm 0.15$	$\mathbf{94.97} \pm 0.23$	$\mathbf{96.97} \pm 0.06$

Table 1: Mean and standard deviation of the test set accuracy for all methods and datasets. *Co* denotes Coauthor.

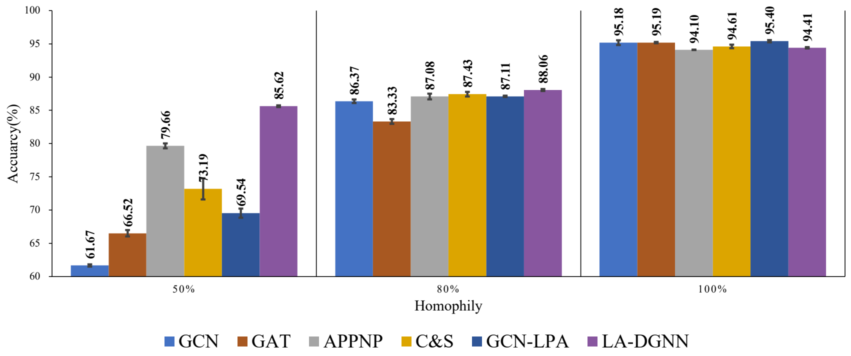

Figure 2: Mean accuracy and standard deviation on Pubmed dataset with different homophily

3.3 Robustness study

Further experiments are conducted to explore the robustness of our model to noisy edges. The homophily of a graph characterizes how likely nodes with the same label are near each other in a graph. Based on Pubmed with 80% homophily, we generate a graph with 50% homophily by adding noisy edges and a graph with 100% homophily by removing all noisy edges. We perform our method and other baselines on the three graphs. Each experiment is conducted three times and the results are shown in Figure 2.

From Figure 2, we observe that the performance of all methods is significantly improved when all noisy edges are removed and they achieve comparable accuracy. When only half of the edges satisfy the label agreement, all baselines experience a large drop while our method still maintains a competitive performance. These experiments demonstrate structure noise has a dramatic effect on model performance and our model is robust to structure noise.

4 Conclusion

In this paper, we propose a decoupled graph neural networks, LA-DGNN, with preprocessing feature propagation and post-processing label propagation. In order to address the structure noise, we train a label agreement model additionally which takes the features of two connected nodes as input to predict the probability they have same labels. The confidence scores of every edge generated by the label agreement model can guide the information propagation across the edges. Aiming to solve the fixed aggregation schema problem, we concatenates intermediate features after each propagation step as a whole to enable our model dynamically adopts importance to neighbors at different distance. Extensive experiments demonstrate LA-DGNN outperforms other classic and SOTA methods on five benchmark datasets and achieves both effectiveness and robustness.

Acknowledgments

The research of Shao-Lun Huang is supported in part by the Shenzhen Science and Technology Program under Grant KQTD20170810150821146, National Key R&D Program of China under Grant 2021YFA0715202 and High-end Foreign Expert Talent Introduction Plan under Grant G2021032013L.

References

Arjun Bakshi, Srinivasan Parthasarathy, and Kannan Srinivasan. Semi-supervised community detection using structure and size. In IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018, pp. 869–874. IEEE Computer Society, 2018. doi: 10.1109/ICDM.2018.00103. URL https://doi.org/10.1109/ICDM.2018.00103.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (eds.), Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020, pp. 639–648. ACM, 2020. doi: 10.1145/3397271.3401063. URL https://doi.org/10.1145/3397271.3401063.

Qian Huang, Horace He, Abhay Singh, Ser-Nam Lim, and Austin R. Benson. Combining label propagation and simple models out-performs graph neural networks. In 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, 2021. URL https://openreview.net/forum?id=8E1-f3VhX1o.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019. URL https://openreview.net/forum?id=H1gL-2A9Ym.

Boris Knyazev, Graham W. Taylor, and Mohamed R. Amer. Understanding attention and generalization in graph neural networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pp. 4204–4214, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Guohao Li, Matthias Müller, Ali K. Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019, pp. 9266–9275. IEEE, 2019. doi: 10.1109/ICCV.2019.00936. URL https://doi.org/10.1109/ICCV.2019.00936.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pp. 3538–3545. AAAI Press, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16098.

Galileo Namata, Ben London, Lise Getoor, Bert Huang, and UMD EDU. Query-driven active surveying for collective classification. In 10th International Workshop on Mining and Learning with Graphs, volume 8, pp. 1, 2012.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. AI Mag., 29(3):93–106, 2008. doi: 10.1609/aimag.v29i3.2157. URL https://doi.org/10.1609/aimag.v29i3.2157.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. CoRR, abs/1811.05868, 2018. URL http://arxiv.org/abs/1811.05868.

Otilia Stretcu, Krishnamurthy Viswanathan, Dana Movshovitz-Attias, Emmanouil A. Platanios, Sujith Ravi, and Andrew Tomkins. Graph agreement models for semi-supervised learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pp. 8710–8720, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/4772c1b987f1f6d8c9d4ef0f3b764f7a-Abstract.html.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Fei Wang and Changshui Zhang. Label propagation through linear neighborhoods. IEEE Trans. Knowl. Data Eng., 20(1):55–67, 2008. doi: 10.1109/TKDE.2007.190672. URL https://doi.org/10.1109/TKDE.2007.190672.

Hongwei Wang and Jure Leskovec. Unifying graph convolutional neural networks and label propagation. CoRR, abs/2002.06755, 2020. URL https://arxiv.org/abs/2002.06755.

Felix Wu, Amauri H. Souza Jr., Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. Simplifying graph convolutional networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, volume 97 of Proceedings of Machine Learning Research, pp. 6861–6871. PMLR, 2019. URL http://proceedings.mlr.press/v97/wu19e.html.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In Jennifer G. Dy and Andreas Krause (eds.), Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, volume 80 of Proceedings of Machine Learning Research, pp. 5449–5458. PMLR, 2018. URL http://proceedings.mlr.press/v80/xu18c.html.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In Sebastian Thrun, Lawrence K. Saul, and Bernhard Schölkopf (eds.), Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada], pp. 321–328. MIT Press, 2003. URL https://proceedings.neurips.cc/paper/2003/hash/87682805257e619d49b8e0dfdc14affa-Abstract.html.

A Structure Noise Influence

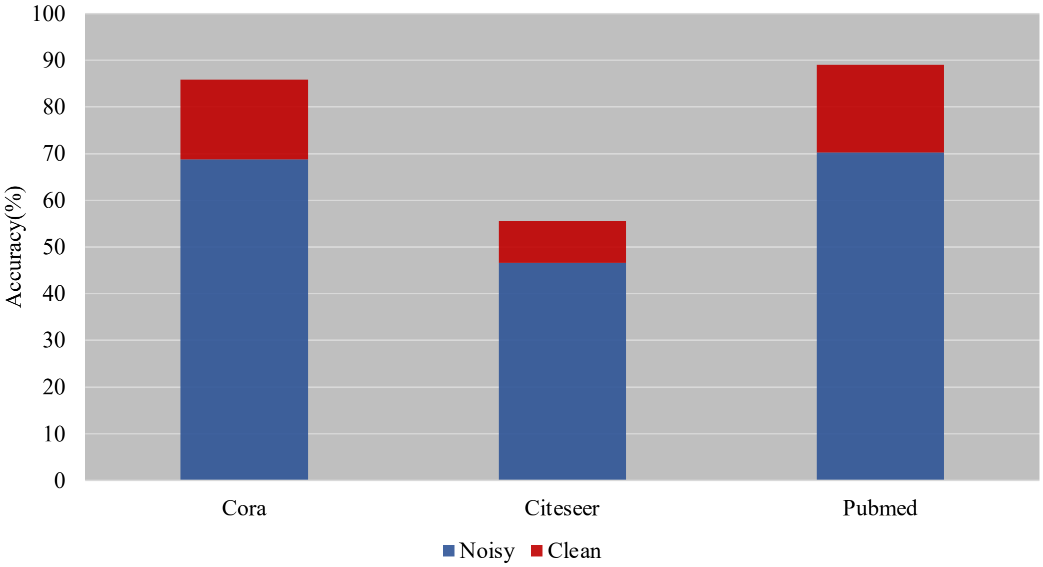

Figure 3: Accuracy of LP on Cora, Citeseer and Pubmed with and without structure noise

We conduct experiments on Cora, Citeseer and Pubmed under the standard split to verify the influence of noisy edges which do not meet label agreement. We apply Label Propagation separately to the original graph and the clear graph with all the noisy edges removed. The classification accuracy is demonstrated in Figure 3. The red areas represent the improvement that the model can obtain on a clean graph compared to a graph containing noisy edges. We observe LP achieves 17.1%, 8.9%, and 18.7% accuracy improvement respectively. The experimental results shows that a significant portion of the potential of the graph learning models is limited by the structure noise.

B Dataset Details

Dataset	Nodes	Edges	Features	Classes	Homophily
Cora	2,708	5,278	1,433	7	81.0%
Citeseer	3,327	4,552	3,703	6	73.6%
Pubmed	19,717	44,324	500	3	80.2%
Coauthor-CS	18,333	81,894	6,805	15	80.8%
Coauthor-Physics	34,493	247,962	8,415	5	93.1%

Table 2: Statistics of the datasets

We select five representative public graph datasets as our benchmark datasets. Their detailed statistics are shown in Table 2. Cora, Citeseer and Pubmed are three most widely used citation networks, where each node represents a paper and an edge indicates a citation relationship. Coauthor-CS and Coauthor-Physics are two co-authorship network with larger size, where each node represents an author and an edge indicates co-author relationship. We randomly split these data sets in the ratio of 60%:20%:20% to form the training set, validation set and test set, which is different from lower label rate settings, in order to ameliorate sensitivity to hyper-parameters.

C Compared methods

In this section, we give a detailed description of baseline methods including both classic and state-of-the-art methods.

- LP (Zhou et al., 2003) Label propagation is a classic semi-supervised learning algorithm that propagates the known labels along the graph to other unlabeled nodes.
- GCN (Kipf & Welling, 2017) Graph Convolutional Network is a widely used approach for semi-supervised learning on graph-structured data that is based on an efficient variant of convolutional neural networks.
- GAT (Velickovic et al., 2018) Graph Attention Networks is a novel convolution-style graph neural networks based on masked self-attentional layers. It specifies fine-grained weights for neighborhood aggregation while does not depend on knowing the entire graph structure upfront.
- JKNet (Xu et al., 2018) Jumping Knowledge Networks is a novel technique to let model flexibly leverages different neighborhood ranges for each node to enable better structure-aware representation. It leverages several approaches to selectively exploit information from neighborhoods of differing locality to break the limit of fixed numbers of neighborhood aggregations.
- APPNP (Klicpera et al., 2019) Approximation Personalized Propagation of Neural Predictions is derived by considering the relationship between GCN and personalized PageRank. It overcomes the limited range problem of many message passing models by decoupling prediction and propagation.
- C&S (Huang et al., 2021) Correct And Smooth only utilizes shallow models that ignore the graph structure with two simple post-processing steps based on label propagation techniques. In C&S, the graph structure is not used to learn parameters but instead as a post-processing mechanism, which reduce magnitude parameters.
- GCN-LPA (Wang & Leskovec, 2020) GCN-LPA is an end-to-end model that unifies GCN and LPA. It serve LPA as regularization to assist the GCN to learn proper edge weights to improve classification performance.

D Implement Details

Here we provide some more details on the models that we use. For all models, we apply the Adam optimizer and tune the learning rate. We use ELU as our activation function and add batch normalization layer after each linear layer. On Cora, Citeseer, the dropout rate is 0.7 while it equals 0.2 on Pubmed, Coauthor-CS, Coauthor-Physics.

For Label Propagation, we propagate 10 steps with a teleport probability of 0.2. For GCN and GAT, we build these models with three convolutional layers and 128 hidden channels. For JKNet, we apply a 3-layer GCN as our base model and use concatenation aggregator. As for APPNP, we use a 2-layer MLP with 10 steps of propagation.

On Cora, Citeseer and Pubmed, the label agreement model and base prediction model are trained jointly with a balancing parameter of 2. On Coauthor-CS and Coauther-Physics, we train label agreement model at first and then train the base predictor based on the predicting edge weights.