DECOUPLED GRAPH NEURAL NETWORKS BASED ON LABEL AGREEMENT MESSAGE PROPAGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Decoupling has become a new paradigm in Graph Neural Networks (GNNs) for its effectiveness and scalability. However, unsatisfying propagation caused by noisy or confused edges can greatly degrade model performance. Fixed propagation schemes also limit the ability of models to learn the importance of a different range of neighbors. To address these problems, we propose a novel decoupled graph model, named LA-DGNN, based on label agreement message propagation and combine the feature after each propagation step as input. In our model 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 concat all features at each propagation step to make the model dynamically learn information about 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 the receptive field, incurring a high computation and memory cost. Furthermore, GNNs incur notorious over-smoothing issues while directly applying multiple layers (Li et al., 2019).

Much recent research has attempted to decouple the feature transformation and neighborhood aggregation in each convolution layer to produce 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. The researchers derived an improved propagation scheme based on personalized PageRank to permit the use of a far greater number of propagation steps without leading to over-smoothing. In contrast, C&S (Huang et al., 2021) propagates label messages; a base predictor is trained with node features that ignores graph structure, and two post operations are applied based on label propagagtion (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 because graph edges are from noisy sources. Therefore, the corresponding labels of some edges are not the same. Additionally, these models treat each propagation equally and use a hyperparameter to mix current information and neighborhood information. However, the importance of nodes varies depending on the distance between neighbors. To address these two problems, based on previous advances, we propose a novel decoupled graph model named LA-DGNN. First, a label agreement model is trained based on the labels in training set to generate a proper confidence for

each edge. The features are then propagated according to the confidence-guided adjacent matrix and we concat each feature for each node after every propagation step. The aggregated features serve as inputs for a base predictor based on multi-layer perceptrons. After training, a post label propagation step is applied to smooth predictions and leverage known labels.

2 METHODOLOGY

The following notations are used in this paper: Let $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ denote 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 $\mathcal{V}^U$.

The model follows the idea of decoupled GNNs, which treat message propagation as pre and post operations. A classifier is trained based on the augmented features after a feature propagation process. Then, a label propagation process is applied to smooth the prediction. In order to ensure the correct propagation paths, a label agreement prediction model is trained based on the known labels. With the aim of achieving the best receptive field for each node, the features are combined into a vector after every propagation step to feed in to the base classifier.

2.1 LABEL AGREEMENT MODEL

Most current 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 LPA as an example; its performance on graphs without noisy edges achieves 17.1%, 8.9%, and 18.7% improvement compared with graphs with noisy edges on Cora, Citeseer, and Pubmed datasets respectively under standard division (see APPENDIX A). Other methods attempt to reduce the impact of the noisy edges. For example, GAT (Velickovic et al., 2018) utilizes the features of two connected nodes to learn the edge weights for convolution operation.

This research takes the concept label agreement from (Stretcu et al., 2019) and separates the edge weight learning process. The features of two neighboring nodes are used as inputs to predict the label consistency of the two nodes. First, the node features are embedded 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, the subtraction and square operations are applied between the two embeddings to eliminate the influence of the order of the two nodes, $\mathbf{e}_{ij} = (\mathbf{e}_i - \mathbf{e}_j)^2$. The calculation of the aggregated embeddings $\mathbf{e}_{ij}$ can be seen 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 with a sigmoid function ϕ to predict the confidence scores that the two nodes have the same label, $s_{ij} = \phi(cls(\mathbf{e}_{ij}))$. The binary cross entropy loss ℓ_{bce} is used as the loss function of the agreement model:

$$\mathcal{L}_{\text{LA}} = \sum_{i \in \mathcal{V}^L, j \in \mathcal{V}^L, ij \in \mathcal{E}} \ell_{bce}(s_{ij}, \mathbb{I}_{y_i = y_j}) \tag{1}$$

2.2 CONFIDENCE-GUIDED FEATURE PROPAGATION

The outputs of the label agreement model reflect the confidence of 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. The label agreement model is used to predict agreement confidences for every edge which serve as the edge weights. Then, the edge weights are used to build the confidence-guided adjacency matrix, $\tilde{\mathbf{A}} = LA - Model(\mathbf{X}, \mathcal{E})$. As before, the symmetric normalization adjacency matrix, $\hat{\mathbf{A}}$, is applied in a 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 $\tilde{\mathbf{A}}$. $\hat{\mathbf{A}}$ is used to construct a K-step feature propagation as,

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

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

After K-step propagation, K-step mixed features are obtained for each node. As discussed previously, different nodes have different sensitivities to different propagation distances. In this model, a simple but effective approach is used, which concats the original features and mixed features after each step as the input. The combination of information at different distances enables the base predictor to dynamically adjust the importance of different receptive field. Propagated features are produced as $\hat{\mathbf{X}} = [\mathbf{X}^{(0)} \| \mathbf{X}^{(1)} \| \ldots \| \mathbf{X}^{(K)}]$.

2.3 Model training

After feature propagation, each node's features are augmented by its neighbors. The augmented features $\hat{\mathbf{X}}$ are used to train a shallow classifier based on multi-layer perceptrons with a softmax function δ applied. The cross entropy measurement ℓ_{ce} is applied as the loss function,

$$\mathcal{L}_{\mathrm{CL}} = \sum_{i \in \mathcal{V}^L} \ell_{ce}(\delta(mlp(\hat{\mathbf{X}}_i)), \mathbf{Y}_i) \tag{3}$$

where $\mathbf{Y}_i$ is the one-hot label of node i. 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 scheme, the total loss is $\mathcal{L} = \mathcal{L}_{\mathrm{CL}} + \beta \mathcal{L}_{\mathrm{LA}}$, 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, a confidence-guided label propagation is also applied to further boost the performance. Each node's label score vector $\mathbf{Z}$ can be obtained by applying a softmax function to the output of the base predictor. The nodes $\mathcal{V}^L$ scores are then replaced with their ground truth to directly use labels, $\mathbf{Z}^L = \mathbf{Y}^L$. A weighted Q-step label propagation based on personalized PageRank is then applied to generate the final prediction scores:

$$\mathbf{Z}^{(k)} = (1-\alpha)\mathbf{Z}^{(0)} + \alpha \hat{\mathbf{A}} \mathbf{Z}^{(k-1)}, \forall k = 1, \ldots, Q \tag{4}$$

where $\mathbf{Z}^{(0)} = \mathbf{Z}$. The classification for node $i \in \mathcal{V}^U$ is $argmax \mathbf{Z}_i^Q$.

3 Experiments

Extensive experiments were conducted to demonstrate the effectiveness of our methods. The quantitative results and analysis of the different evaluations are presented here.

3.1 Datasets and baselines

Five public datasets were used for evaluation: Cora (Sen et al., 2008), Citeseer (Sen et al., 2008), and Pubmed (Namata et al., 2012), which are three classic citation network graphs; as well as Coauthor-CS (Shchur et al., 2018) and Coauthor-Physics (Shchur et al., 2018), which are two co-authorship graphs based on the Microsoft Academic Graph. For the train/valid/test splits, random splits of 60%/20%/20% were used, following (Wang & Leskovec, 2020).

In the experiments, our model is compared 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), and GCN-LPA (Wang & Leskovec, 2020). LP is a parameter-free method, while GCN, GAT and JKNet are classic graph models. APPNP propagates features at first, while C&S propagates 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 result and standard deviation are given in Table 1.

	Cora	Citeseer	Pubmed	Coauthor-CS	Coauthor-Physics
LP	86.69	71.88	80.5	91.54	95.58
GCN	87.91 ± 0.46	74.89 ± 0.49	86.37 ± 0.27	92.58 ± 0.23	96.14 ± 0.1
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

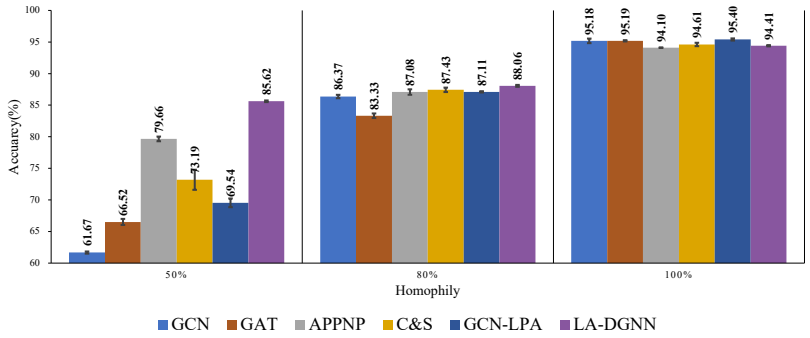

Figure 1: Accuracy on Pubmed dataset with different homophily

There are several observations from the results tabel 1. Our model outperforms all of the baseline methods on each dataset under the semi-supervised setting. For example, our model improves on the best-performing baseline model GCN-LPA, and by an absolute 0.84% accuracy on Cora. For large graph Coauthor-CS and Coauthor-Physics, the average accuracies are also improved. Although LP is parameter free, it still demonstrates competitive performance on most datasets. GCN and GAT produced worse results for their limited and fixed receptive field. JKNet and APPNP outperform these shallow GNNs for its access each propagated-step features or large receptive field. C&S and GCN-LPA also achieve a higher performance benefit from the direct use of label information.

3.3 Robustness Study

Further experiments were conducted to explore robustness of our model to noisy edges. The homophily of a graph characterizes how likely it is that nodes with the same label are near each other in a graph. Compared to original Pubmed, with 80% homophily, we were able to generate a graph with 50% homophily by adding noisy edges and a graph with 100% homophily by removing all noisy edges. The mean accuracy and standard deviation on them are in Figure 2. It can be seen that the performance of all methods is significantly improved when all noisy edges are removed. When only half of the edges satisfy the label agreement, all of the other methods experience a large drop, while our method maintains a competitive performance. These experiments indicate that structure noise has a dramatic effect on model performance, and that our model is robust to structure noise.

4 Conclusion

In this paper, we propose a decoupled graph neural network LA-DGNN with a pre-processing feature propagation and a post-processing label propagation, whose propagations are confidence-guided. The confidence is generated by a label agreement model to address the edge noise, which takes the features of two connected nodes as inputs to predict the probability that they have the same labels. We also performed a concat on the features after each propagation step to enable our model to dynamically adopt the importance to neighbors at different distances. Extensive experiments demonstrate that our model outperforms other representative methods on five benchmark datasets.

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

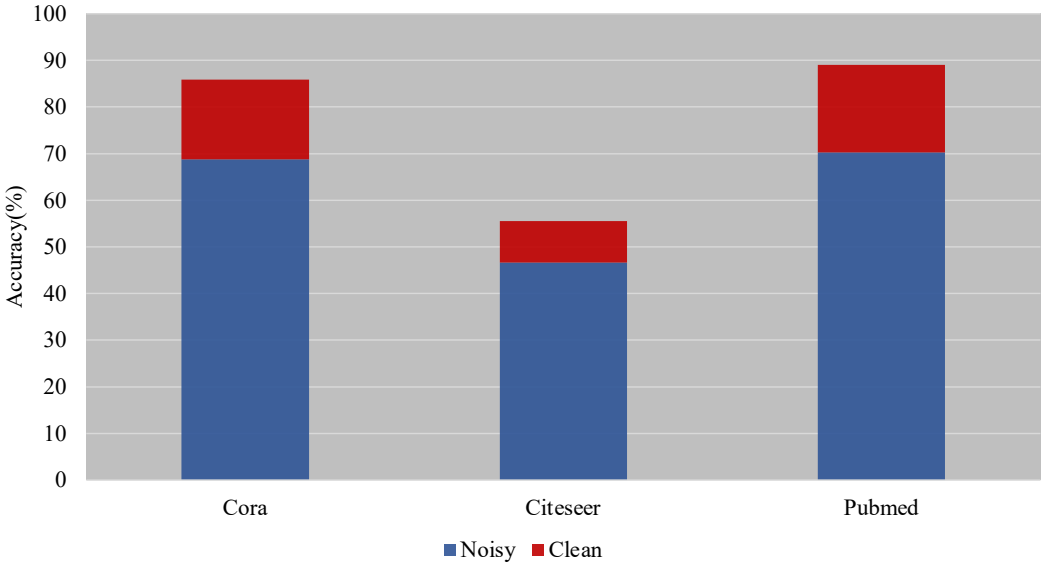

Figure 2: Accuracy of LP on Cora, Citeseer and Pubmed with and without structure noise

Experiments were carried out on Cora, Citeseer, and Pubmed under the standard split to verify the influence of noisy edges that do not meet label agreement. Label propagation is applied separately on the original graph and on the graph with all the noisy edges removed. The results are shown in Figure 2. The red areas represent the improvement that the model can obtain on a clean graph compared to a graph containing noisy edges. The experimental results show 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

Five representative public graph datasets were selected as the benchmark datasets and 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 larger co-authorship networks, where each node represents an author and an edge indicates a co-author relationship. These datasets were randomly split in the ratio of 60%:20%:20% to form the training set, validation set and test set; this is different from lower label rate settings, in order to ameliorate sensitivity to hyperparameters.

C COMPARED METHODS

This section gives a detailed description of the 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 Network is a novel convolution-style graph neural networks based on masked self-attentional layers. It specifies fine-grained weights for neighborhood aggregation, while it does not depend on knowing the entire graph structure up front.
- **JKNet** (Xu et al., 2018) Jumping Knowledge Network is a novel technique that allows models to flexibly leverage 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 uses 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 is instead used as a post-processing mechanism, which reduces magnitude parameters.
- **GCN-LPA** (Wang & Leskovec, 2020) GCN-LPA is an end-to-end model that unifies GCN and LPA. It serves LPA as a regularization to assist the GCN to learn proper edge weights to improve classification performance.

D IMPLEMENT DETAILS

This section provides further details on the models used. For all models the Adam optimizer was applied to tune the learning rate. ELU was used as the activation function and a batch normalization layer was added after each linear layer. On Cora and Citeseer the dropout rate was 0.7, while on Pubmed, Coauthor-CS, and Coauthor-Physics it was 0.2.

For Label Propagation, the method was to propagate 10 steps with a teleport probability of 0.2. The GCN and GAT models were built with three convolutional layers and 128 hidden channels. For JKNet, a 3-layer GCN was applied as the base model and a concatenation aggregator was used. For APPNP a 2-layer multi-layer perceptrons with 10 steps of propagation is used.

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 the label agreement model was trained first and then the base predictor was trained based on the predicting edge weights.