REFERENCES

Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In ICML, volume 97 of Proceedings of Machine Learning Research, pp. 21–29. PMLR, 2019.

Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In AAAI, pp. 3950–3957. AAAI Press, 2021.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In ICLR. OpenReview.net, 2022.

A. M. Buchanan and Andrew W. Fitzgibbon. Damped newton algorithms for matrix factorization with missing data. In CVPR (2), pp. 316–322. IEEE Computer Society, 2005.

Ricardo Cabral, Fernando De La Torre, Joao P. Costeira, and Alexandre Bernardino. Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), December 2013.

Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. Commun. ACM, 55(6):111–119, 2012.

Dorwin Cartwright and Frank Harary. Structural balance: a generalization of heider's theory. Psychological review, 63(5):277, 1956.

Pei Chen. Optimization algorithms on subspaces: Revisiting missing data problem in low-rank matrix. Int. J. Comput. Vis., 80(1):125–142, 2008.

Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In ICLR. OpenReview.net, 2021.

Mark A. Davenport and Justin K. Romberg. An overview of low-rank matrix recovery from incomplete observations. IEEE J. Sel. Top. Signal Process., 10(4):608–622, 2016.

James A Davis. Clustering and structural balance in graphs. Human relations, 20(2):181–187, 1967.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. arXiv preprint arXiv:1903.02428, 2019.

Kimon Fountoulakis, Amit Levi, Shenghao Yang, Aseem Baranwal, and Aukosh Jagannath. Graph attention retrospective. CoRR, abs/2202.13060, 2022.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In NIPS, pp. 1024–1034, 2017.

Trevor Hastie, Rahul Mazumder, Jason D. Lee, and Reza Zadeh. Matrix completion and low-rank SVD via fast alternating least squares. J. Mach. Learn. Res., 16:3367–3402, 2015.

Roger A. Horn and Charles R. Johnson. Topics in Matrix Analysis. Cambridge University Press, 1991.

Cho-Jui Hsieh, Kai-Yang Chiang, and Inderjit S. Dhillon. Low rank modeling of signed networks. In KDD, pp. 507–515. ACM, 2012.

Du Q. Huynh, R. Hartley, and Anders Heyden. Outlier correction in image sequences for the affine camera. In ICCV, pp. 585–590. IEEE Computer Society, 2003.

Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining, pp. 66–74, New York, NY, USA, 2020. Association for Computing Machinery.

Dongkwan Kim and Alice Oh. How to find your friendly neighborhood: Graph attention design with self-supervision. In ICLR. OpenReview.net, 2021.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In ICLR (Poster). OpenReview.net, 2017.

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In ICLR (Poster). OpenReview.net, 2019.

Xiang Li, Renyu Zhu, Yao Cheng, Caihua Shan, Siqiang Luo, Dongsheng Li, and Weining Qian. Finding global homophily in graph neural networks when meeting heterophily. In ICML, volume 162 of Proceedings of Machine Learning Research, pp. 13242–13256. PMLR, 2022.

Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser-Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. In NeurIPS, pp. 20887–20902, 2021.

Guangcan Liu and Shuicheng Yan. Latent low-rank representation for subspace segmentation and feature extraction. In ICCV, pp. 1615–1622. IEEE Computer Society, 2011.

Guangcan Liu, Zhouchen Lin, and Yong Yu. Robust subspace segmentation by low-rank representation. In ICML, pp. 663–670. Omnipress, 2010.

Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Is heterophily A real nightmare for graph neural networks to do node classification? CoRR, abs/2109.05641, 2021.

Deyu Meng and Fernando De La Torre. Robust matrix factorization with unknown noise. In Proceedings of the IEEE International Conference on Computer Vision, pp. 1337–1344, 2013.

Takayuki Okatani and Koichiro Deguchi. On the wiberg algorithm for matrix factorization in the presence of missing components. Int. J. Comput. Vis., 72(3):329–337, 2007.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In ICLR. OpenReview.net, 2020.

Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. J. Complex Networks, 9(2), 2021.

Heung-Yeung Shum, Katsushi Ikeuchi, and Raj Reddy. Principal component analysis with missing data and its application to polyhedral object modeling. IEEE Trans. Pattern Anal. Mach. Intell., 17(9):854–867, 1995.

Susheel Suresh, Vinith Budde, Jennifer Neville, Pan Li, and Jianzhu Ma. Breaking the limit of graph neural networks by improving the assortativity of graphs with local mixing patterns. In KDD, pp. 1541–1551. ACM, 2021.

Jiliang Tang, Xia Hu, Huiji Gao, and Huan Liu. Exploiting local and global social context for recommendation. In IJCAI, pp. 2712–2718. IJCAI/AAAI, 2013.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In ICLR (Poster). OpenReview.net, 2018.

T Wiberg. Computation of principal components when data are missing. In Proc. of Second Symp. Computational Statistics, pp. 229–236, 1976.

Yujun Yan, Milad Hashemi, Kevin Swersky, Yaoqing Yang, and Danai Koutra. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. CoRR, abs/2102.06462, 2021.

Liang Yang, Mengzhe Li, Liyang Liu, Bingxin Niu, Chuan Wang, Xiaochun Cao, and Yuanfang Guo. Diverse message passing for attribute with heterophily. In NeurIPS, pp. 4751–4763, 2021.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In ICML, volume 48 of JMLR Workshop and Conference Proceedings, pp. 40–48. JMLR.org, 2016.

Xin Zheng, Yixin Liu, Shirui Pan, Miao Zhang, Di Jin, and Philip S. Yu. Graph neural networks for graphs with heterophily: A survey. CoRR, abs/2202.07082, 2022.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In NeurIPS, 2020.

Jiong Zhu, Ryan A. Rossi, Anup Rao, Tung Mai, Nedim Lipka, Nesreen K. Ahmed, and Danai Koutra. Graph neural networks with heterophily. In AAAI, pp. 11168–11176. AAAI Press, 2021.

A Appendix

A.1 Surrogate function

By definition, we have $\mathrm{P}_\Omega(\mathbf{M}_{i,j}) = \mathbf{M}_{i,j}$ if $\mathrm{P}_\phi(\mathbf{M}_{i,j}) = 0$ and $\mathrm{P}_\Omega(\mathbf{M}_{i,j}) = 0$ if $\mathrm{P}_\phi(\mathbf{M}_{i,j}) = \mathbf{M}_{i,j}$, $\forall (i,j) \in \mathcal{V} \times \mathcal{V}$. This leads to the following equation

$$\begin{aligned} &\|\mathrm{P}_\Omega(\mathbf{Z}_U \bar{\mathbf{V}}^{(l)^T} - \tilde{\mathbf{A}}) + \mathrm{P}_\phi(\mathbf{Z}_U \bar{\mathbf{V}}^{(l)^T} - \bar{\mathbf{U}}^{(l)} \bar{\mathbf{V}}^{(l)^T})\|_F^2 = \\ &\|\mathrm{P}_\Omega(\tilde{\mathbf{A}} - \mathbf{Z}_U \bar{\mathbf{V}}^{(l)^T}) + \mathrm{P}_\phi(\bar{\mathbf{U}}^{(l)} \bar{\mathbf{V}}^{(l)^T} - \mathbf{Z}_U \bar{\mathbf{V}}^{(l)^T})\|_F^2 = \|P_\Omega(\tilde{\mathbf{A}}) + P_\phi(\bar{\mathbf{U}}^{(l)} \bar{\mathbf{V}}^{(l)^T}) - \mathbf{Z}_U \bar{\mathbf{V}}^{(l)^T}\|_F^2, \end{aligned} \tag{21}$$

wherein $\mathbf{Z}_U$ gets rid of the element-wise function, thus we can directly derive the closed-form solution.

A.2 Pseudocodes

We summarize the pseudocodes of LRGNN as follows.

Algorithm 1 LRGNN

Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, node features $\mathbf{X}$, adjacencay matrix $\mathbf{A}$, Pseudo labels $\bar{\mathbf{Y}}$
Output: The node representation matrix of the last layer of LRGNN

```
Calculate signed adjacency matrix using Pseudo labels.
Calculate H^(0) using Eq. (5)
for l = 0 to L − 1 do
    Initialize V_init^(l) and U_init^(l) using Eq. (14)
    for k = 0 to K − 1 do
        Update U^(l) using Eq. (10)
        Update V^(l) using Eq. (12)
    end for
    Calculate H^(l+1) using Eq. (6)
end for
return H^(L)
```

A.3 Time complexity

Directly solving the inverse of $\gamma \mathbf{I}_n + (1-\beta)^2 \mathbf{H}^{(l)} \mathbf{H}^{(l)^T}$ takes time complexity of $O(n^3)$. Using the well-known Woodbury formula, the inverse can be equivalently computed as

$$[\gamma \mathbf{I}_n + (1-\beta)^2 \mathbf{H}^{(l)} \mathbf{H}^{(l)^T}]^{-1} = \frac{1}{\gamma} \mathbf{I}_n - \frac{(1-\beta)^2}{\gamma^2} \mathbf{H}^{(l)} [\mathbf{I}_c + \frac{(1-\beta)^2}{\gamma} \mathbf{H}^{(l)^T} \mathbf{H}^{(l)}]^{-1} \mathbf{H}^{(l)^T} \tag{22}$$

The time complexity for solving $[\mathbf{I}_c + \frac{(1-\beta)^2}{\gamma} \mathbf{H}^{(l)^T} \mathbf{H}^{(l)}]^{-1}$ is only $O(c^3)$, where c is the number of classes and a very small number in general. Further, we always perform the matrix multiplication in a right-to-left way, which substantially reduces the time cost. For example, the time complexity of $(\mathbf{H}^{(l)} \mathbf{H}^{(l)^T}) \tilde{\mathbf{U}}^{(l)}$ is $O(n^2 c)$ while the time complexity of $\mathbf{H}^{(l)} (\mathbf{H}^{(l)^T} \tilde{\mathbf{U}}^{(l)})$ is only $O(nc^2)$. Although $\widehat{\mathbf{A}}$ is not sparse, $\widehat{\mathbf{A}} \bar{\mathbf{V}}^{(l)}$ can be decomposed into sparse-dense matrix multiplications, resulting in time complexity of $O(mq + nq^2)$, where m denotes the number of edges.

We first consider the cost of updating $\mathbf{U}$.

$$\tilde{\mathbf{U}}^{(l)} = [\gamma\widehat{\mathbf{A}}\bar{\mathbf{V}}^{(l)} + \delta\mathbf{H}^{(l)}\mathbf{H}^{(l)^T}\bar{\mathbf{V}}^{(l)} - \beta\delta\mathbf{H}^{(0)}\mathbf{H}^{(l)^T}\bar{\mathbf{V}}^{(l)}][\gamma\bar{\mathbf{V}}^{(l)^T}\bar{\mathbf{V}}^{(l)} + \delta^2\bar{\mathbf{V}}^{(l)^T}\mathbf{H}^{(l)}\mathbf{H}^{(l)^T}\bar{\mathbf{V}}^{(l)}]^{-1}, \tag{23}$$

where $\delta = 1 - \beta$. $\bar{\mathbf{V}}$ and $\mathbf{H}^{(l)}$ are matrices of size $n \times q$ and size $n \times c$, respectively. The time complexity of the following computing order is $O(mq + q^3 + ncq + nq^2)$

$$\tilde{\mathbf{U}}^{(l)} = [\gamma\widehat{\mathbf{A}}\bar{\mathbf{V}}^{(l)} + \delta\mathbf{H}^{(l)}(\mathbf{H}^{(l)^T}\bar{\mathbf{V}}^{(l)}) - \beta\delta\mathbf{H}^{(0)}(\mathbf{H}^{(l)^T}\bar{\mathbf{V}}^{(l)})][\gamma\bar{\mathbf{V}}^{(l)^T}\bar{\mathbf{V}}^{(l)} + \delta^2(\bar{\mathbf{V}}^{(l)^T}\mathbf{H}^{(l)})(\mathbf{H}^{(l)^T}\bar{\mathbf{V}}^{(l)})]^{-1}, \tag{24}$$

where $\widehat{\mathbf{A}}\bar{\mathbf{V}}^{(l)}$ can be realized by sparse-dense matrix multiplications.

$$\begin{aligned}\widehat{\mathbf{A}}\bar{\mathbf{V}}^{(l)} &= (\mathrm{P}_\Omega(\tilde{\mathbf{A}}) + P_\phi(\bar{\mathbf{U}}^{(l)}\bar{\mathbf{V}}^{(l)^T}))\bar{\mathbf{V}}^{(l)} = (\mathrm{P}_\Omega(\tilde{\mathbf{A}}) + \bar{\mathbf{U}}^{(l)}\bar{\mathbf{V}}^{(l)^T} - P_\Omega(\bar{\mathbf{U}}^{(l)}\bar{\mathbf{V}}^{(l)^T}))\bar{\mathbf{V}}^{(l)} \\ &= \mathrm{P}_\Omega(\tilde{\mathbf{A}})\bar{\mathbf{V}}^{(l)} + \bar{\mathbf{U}}^{(l)}\bar{\mathbf{V}}^{(l)^T}\bar{\mathbf{V}}^{(l)} - P_\Omega(\bar{\mathbf{U}}^{(l)}\bar{\mathbf{V}}^{(l)^T})\bar{\mathbf{V}}^{(l)},\end{aligned} \tag{25}$$

where $\mathrm{P}_\Omega(\tilde{\mathbf{A}})\bar{\mathbf{V}}^{(l)}$ and $P_\Omega(\bar{\mathbf{U}}^{(l)}\bar{\mathbf{V}}^{(l)^T})\bar{\mathbf{V}}^{(l)}$ are sparse-dense matrix multiplications. the time complexity of $\bar{\mathbf{U}}^{(l)}(\bar{\mathbf{V}}^{(l)^T}\bar{\mathbf{V}}^{(l)})$ is $O(nq^2)$.

Now we consider the calculating of $\tilde{\mathbf{V}}^{(l)}$

$$\begin{aligned}\tilde{\mathbf{V}}^{(l)} = [\gamma\mathbf{I}_n + (1-\beta)^2\mathbf{H}^{(l)}\mathbf{H}^{(l)^T}]^{-1}[\gamma\widehat{\mathbf{A}}^T\tilde{\mathbf{U}}^{(l)} + (1-\beta)\mathbf{H}^{(l)}\mathbf{H}^{(l)^T}\tilde{\mathbf{U}}^{(l)} - \beta(1-\beta)\mathbf{H}^{(l)}\mathbf{H}^{(0)^T}\tilde{\mathbf{U}}^{(l)}]\cdot \\ [\tilde{\mathbf{U}}^{(l)^T}\tilde{\mathbf{U}}^{(l)}]^{-1}\end{aligned} \tag{26}$$

The time complexity of computing $\gamma\widehat{\mathbf{A}}^T\tilde{\mathbf{U}}^{(l)} + (1-\beta)\mathbf{H}^{(l)}\mathbf{H}^{(l)^T}\tilde{\mathbf{U}}^{(l)} - \beta(1-\beta)\mathbf{H}^{(l)}\mathbf{H}^{(0)^T}\tilde{\mathbf{U}}^{(l)}$ is $O(mq + q^3 + ncq + nq^2)$, which can be achieved by reordering the matrix multiplications and decomposing $\widehat{\mathbf{A}}^T$ into sparse matrices using the same tricks as mentioned above. Note that

$$[\gamma\mathbf{I}_n + (1-\beta)^2\mathbf{H}^{(l)}\mathbf{H}^{(l)^T}]^{-1} = \frac{1}{\gamma}\mathbf{I}_n - \frac{(1-\beta)^2}{\gamma^2}\mathbf{H}^{(l)}[\mathbf{I}_c + \frac{(1-\beta)^2}{\gamma}\mathbf{H}^{(l)^T}\mathbf{H}^{(l)}]^{-1}\mathbf{H}^{(l)^T} \tag{27}$$

Denote $[\gamma\widehat{\mathbf{A}}^T\tilde{\mathbf{U}}^{(l)} + (1-\beta)\mathbf{H}^{(l)}\mathbf{H}^{(l)^T}\tilde{\mathbf{U}}^{(l)} - \beta(1-\beta)\mathbf{H}^{(l)}\mathbf{H}^{(0)^T}\tilde{\mathbf{U}}^{(l)}][\tilde{\mathbf{U}}^{(l)^T}\tilde{\mathbf{U}}^{(l)}]^{-1}$ by $\mathbf{Q}$, which is a matrix of size $n \times q$. We have

$$([\gamma\mathbf{I}_n + (1-\beta)^2\mathbf{H}^{(l)}\mathbf{H}^{(l)^T}]^{-1})^T\mathbf{Q} = \frac{1}{\gamma}\mathbf{Q} - \frac{(1-\beta)^2}{\gamma^2}\mathbf{H}^{(l)}([\mathbf{I}_c + \frac{(1-\beta)^2}{\gamma}\mathbf{H}^{(l)^T}\mathbf{H}^{(l)}]^{-1})^T\mathbf{H}^{(l)^T}\mathbf{Q} \tag{28}$$

The time complexity of (28) is $O(nc^2 + c^3 + ncq)$ when performing the matrix multiplications in a right-to-left way. Since the size of $\tilde{\mathbf{V}}^{(l)^T}\mathbf{H}^{(l)}$ is $q \times c$, $\tilde{\mathbf{U}}^{(l)}(\tilde{\mathbf{V}}^{(l)^T}\mathbf{H}^{(l)})$ can be performed in $O(ncq)$.

In addition to the multiplication reordering, we can avoid the repeated calculations of the common subexpressions. For example, $\mathbf{H}^{(l)^T}\bar{\mathbf{V}}^{(l)}$ and $\mathbf{H}^{(l)}\mathbf{H}^{(l)^T}\bar{\mathbf{V}}^{(l)}$ in the computing of $\tilde{\mathbf{U}}^{(l)}$.

In general, given that q and c are very small numbers and $m \gg n$, the time complexity is bounded by $O(dmq)$, where d is a constant and $d \ll n$.

A.4 Related work

In this section, we discuss relevant work that addresses the heterophily challenging. Abu-El-Haija et al. (2019) acknowledges the limitations of current GNNs in learning on graphs with heterophily and proposes to exploit higher-order information by aggregating multi-hop neighborhoods. The authors of Zhu et al. (2020) further identified several effective designs and provided theoretical justifications. Chien et al. (2021) generalizes the PageRank and proposes GPR-GNN that performs well under heterophily. FAGCN (Bo et al., 2021) utilizes a self-gating attention mechanism to adaptively learn the proportion of low-frequency and high-frequency signals. Later, WRGAT (Suresh et al., 2021) transforms the original graph into a new multi-relational one with a higher homophily ratio. The authors of Yan et al. (2021) regard oversmoothing and heterophily as two sides of the same coin. They suggest addressing these two issues via degree correction and signed message. LINKX (Lim et al., 2021) first embeds node features and graph topology separately and then combines them with MLPs. Recently, GloGNN (Li et al., 2022) proposes to leverage global homophily and derives a coefficient matrix that optimizes a well-designed objective function. Zheng et al. (2022) provides a comprehensive survey on GNNs for heterophilious graphs.

	Texas	Wisconsin	Cornell	Actor	Squirrel	Chameleon	Cora	Citeseer	Pubmed
Edge Hom.	0.11	0.21	0.30	0.22	0.22	0.23	0.81	0.74	0.80
#Nodes	2,708	251	183	7,600	5,201	2,277	2,708	3,327	19,717
Avg. Deg.	1.61	1.86	1.53	3.52	38.16	13.80	1.95	1.40	2.25
#Features	1,703	1,703	1,703	931	2,089	2,325	1,433	3,703	500
#Classes	5	5	5	5	5	5	6	7	3

Table 3: Statistics of the datasets used.

A.5 DATASETS

We here briefly introduce the datasets used in our experiments. In particular, these datasets span various domains and edge homophily.

Cora, Citeseer and Pubmed are citation networks where nodes represent scientific papers and edges are citation relationships. Node features are bag-of-words representations and each label represents the field that the paper belongs to.

Actor is a co-occurrence network generated from the film-director-actor-writer network, where node features are bag-of-words representations of the Wikipedia pages of actors. Edges symbolize the two actors' co-occurrence on the same web page.

Cornell, Texas and Wisconsin are collected as part of CMU WebKB project. In these datasets, nodes are university web pages and edges are hyperlinks between these pages.

Chameleon and Squirrel are two networks of web pages on Wikipedia regarding animals. Node features are bag-of-words representations of nouns in the respective pages. The task is to classify pages into five categories based on the average traffic they received.

Synthetic graphs are controlled by the node-level homophily ratio and the average degree. Specifically, a random graph contains n nodes per class and c classes, with two probabilities p_{in} and p_{out}, where p_{in} corresponds to the probability of forming a intra-class edge, and p_{out} corresponds to the probability of forming a inter-class edge. We choose p_{in} and p_{out} by $p_{in} + (c-1) \cdot p_{out} = \delta$, and the average degree of the random graph is $d_{avg} = n\delta$. We set n to 500, c to 5, and select d_{avg} from $\{0.5, 5, 20\}$, p_{in} from $\{0.1\delta, 0.3\delta, 0.5\delta, 0.7\delta, 0.9\delta\}$. The node features are sampled from Gaussian distributions where the centers of clusters are vertices of a hypercube. Nodes are randomly split into $(10\%/45\%/45\%)$ for training/validation/testing. Note that $p_{in} = 0.9\delta$ indicates strong homophily and $p_{in} = 0.1\delta$ corresponds to strong heterophily.

A.6 NUMERICAL SIMULATIONS

We conduct numerical simulation experiments to study the performance of LRR on subspace clustering. Specifically, we are interested in whether solving the optimization problem (19) can give a low-rank representation that reveals the membership of nodes. Denote by $\mathbf{U}_*^{(l)}$ and $\mathbf{V}_*^{(l)}$ of size $n \times k$ that minimize the following objective function *i.e.*,(19)

$$\|\mathbf{H}^{(l)} - (1-\beta)\mathbf{U}^{(l)}{\mathbf{V}^{(l)}}^T\mathbf{H}^{(l)} - \beta\mathbf{H}^{(0)}\|_F^2 + \lambda\|\mathbf{U}^{(l)}{\mathbf{V}^{(l)}}^T\|_F^2 \tag{29}$$

Here, we specify $l = 0$ and $l = 2$. We randomly generate 5 independent clusters as most real-world datasets used in experiments have 5 classes. The rank of each subspace is $d/5$ where d is the dimension, a parameter to be specified. For each subspace S_i, we randomly sample n_i vectors with n_i a random variable ranging from 20 to 50. We report the average values of the within-subspace and between-subspace elements w.r.t. different parameters d, k and λ. The results are shown in Figure 5. We can make three conclusions from it. (1) The larger the dimension is, the better the representation reveals the membership. (2) When the dimension is small, the effect of $k = 5$ is better than that of $k = 10$ while as the dimension increases, the larger the k, the better the effect is. This implies that k should be set to fit the rank of subspaces. (3) Increasing λ consistently decreases the mean of within-subspace elements. Therefore, in our objective function (7), we exclude the regularization term. We can also see from the figure that low-rank representations do reveal the membership: within-subspace elements are dense, and between-subspace elements are sparse.

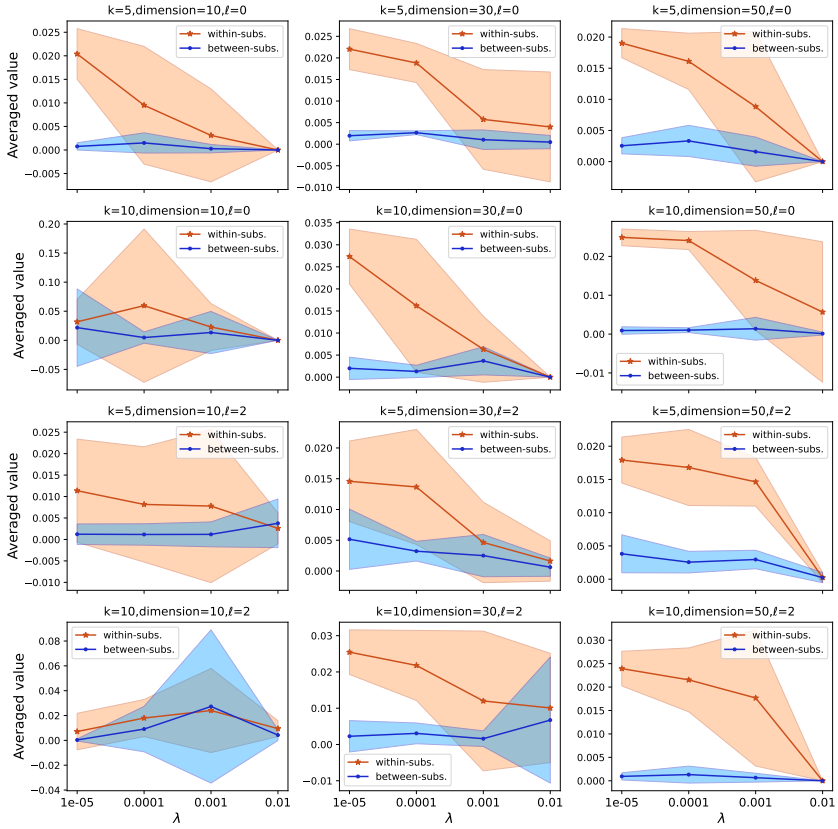

Figure 5: Numerical simulation results. x-axis corresponds to λ. y-axis corresponds to the average values of the within-subspace/between-subspace elements. The shaded regions correspond to 95% confidence intervals. A more significant difference between within-subspace and between-subspace suggests the representation better represents the membership.

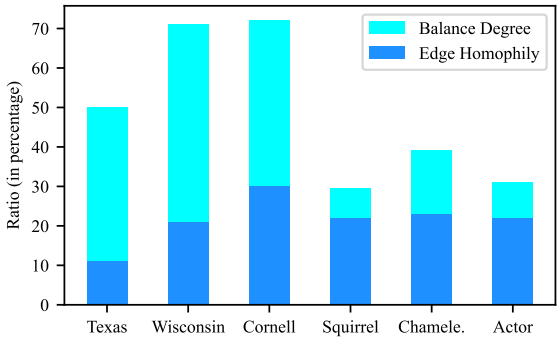

Figure 6: Balance degree v.s. edge homophily.

A.7 BALANCE DEGREE

We here investigate the balance degree of the heterophilious datasets. The balance theory includes four statements: (1) An enemy of my friend is my enemy; (2) A friend of my friend is my friend; (3) A friend of my enemy is my enemy; (4) An enemy of my enemy is my friend. Since the first three are always true, we want to know to what extent the fourth statement holds, which is described by

$B_{\mathcal{G}}$ defined as

$$B_{\mathcal{G}} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \frac{|\{k \in \mathcal{N}(u) : y_k = y_v, y_k \neq y_u | u \in \mathcal{N}(v) : y_u \neq y_v\}|}{|\{k \in \mathcal{N}(u) : y_k \neq y_u | u \in \mathcal{N}(v) : y_u \neq y_v\}|} \tag{30}$$

It is clear that $B_{\mathcal{G}} \in [0, 1]$ and it describes the homophily degree of the enemy's enemies. To better reveal to what extent the fourth statement holds, we utilize the edge homophily for comparison, which is defined as

$$\frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \frac{|\{(u, v) \in \mathcal{E} : y_u = y_v\}|}{|\mathcal{E}|} \tag{31}$$

The statistics are shown in Figure 6. Although the weak balance theory does not assume that the enemies of an enemy are friends, the homophily ratio of the enemy's enemies is higher than that of the direct neighbor. This phenomenon implies that the implicit meaning of the link relationship is the 2-order affinity.

A.8 PROOF

Theorem 3. *(Correctness) The objective function (7) is nonincreasing under the update rules (10) and (12),*

$$F(\tilde{\mathbf{U}}^{(l)}, \tilde{\mathbf{V}}^{(l)}) \leq F(\tilde{\mathbf{U}}^{(l)}, \bar{\mathbf{V}}^{(l)}) \leq F(\bar{\mathbf{U}}^{(l)}, \bar{\mathbf{V}}^{(l)}) \tag{32}$$

Proof. The design of the surrogate function follows the definition of the Majorization-Minimization (MM) algorithm. An MM algorithm operates by defining a surrogate function that minorizes the objective function. We begin with presenting the following two observations:

$$\mathrm{S}_U(\mathbf{U}|\mathbf{U}, \mathbf{V}) = F(\mathbf{U}, \mathbf{V}), \mathrm{S}_V(\mathbf{V}|\mathbf{U}, \mathbf{V}) = F(\mathbf{U}, \mathbf{V}) \tag{33}$$

and

$$\mathrm{S}_U(\mathbf{Z}_U|\mathbf{U}, \mathbf{V}) \geq F(\mathbf{Z}_U, \mathbf{V}), \mathrm{S}_V(\mathbf{Z}_V|\mathbf{U}, \mathbf{V}) \geq F(\mathbf{U}, \mathbf{Z}_V) \tag{34}$$

The first equation is easy to derived.

$$\begin{aligned} \mathrm{S}_U(\mathbf{U}|\mathbf{U}, \mathbf{V}) &= \|\mathbf{H}^{(l)} - (1-\beta)\mathbf{U}\mathbf{V}\mathbf{H}^{(l)} - \beta\mathbf{H}^{(0)}\|_F^2 + \gamma\|\mathrm{P}_\Omega(\mathbf{U}\mathbf{V} - \tilde{\mathbf{A}}) + \mathrm{P}_\phi(\mathbf{U}\mathbf{V} - \mathbf{U}\mathbf{V})\|_F^2 \\ &= \|\mathbf{H}^{(l)} - (1-\beta)\mathbf{U}\mathbf{V}\mathbf{H}^{(l)} - \beta\mathbf{H}^{(0)}\|_F^2 + \gamma\|\mathrm{P}_\Omega(\mathbf{U}\mathbf{V} - \tilde{\mathbf{A}})\|_F^2 = F(\mathbf{U}, \mathbf{V}) \end{aligned} \tag{35}$$

$$\begin{aligned} \mathrm{S}_V(\mathbf{V}|\mathbf{U}, \mathbf{V}) &= \|\mathbf{H}^{(l)} - (1-\beta)\mathbf{U}\mathbf{V}\mathbf{H}^{(l)} - \beta\mathbf{H}^{(0)}\|_F^2 + \gamma\|\mathrm{P}_\Omega(\mathbf{U}\mathbf{V} - \tilde{\mathbf{A}}) + \mathrm{P}_\phi(\mathbf{U}\mathbf{V} - \mathbf{U}\mathbf{V})\|_F^2 + \\ &= \|\mathbf{H}^{(l)} - (1-\beta)\mathbf{U}\mathbf{V}\mathbf{H}^{(l)} - \beta\mathbf{H}^{(0)}\|_F^2 + \gamma\|\mathrm{P}_\Omega(\mathbf{U}\mathbf{V} - \tilde{\mathbf{A}})\|_F^2 = F(\mathbf{U}, \mathbf{V}) \end{aligned} \tag{36}$$

By definition, we have $\mathrm{P}_\Omega(\mathbf{M}_{i,j}) = \mathbf{M}_{i,j}$ if $\mathrm{P}_\phi(\mathbf{M}_{i,j}) = 0$ and $\mathrm{P}_\Omega(\mathbf{M}_{i,j}) = 0$ if $\mathrm{P}_\phi(\mathbf{M}_{i,j}) = \mathbf{M}_{i,j}$, for $\forall(i, j) \in \mathcal{V} \times \mathcal{V}$. Using this fact, we can rewrite $\|\mathrm{P}_\Omega(\cdot) + \mathrm{P}_\phi(\cdot)\|_F^2$ as $\|\mathrm{P}_\Omega(\cdot)\|_F^2 + \|\mathrm{P}_\phi(\cdot)\|_F^2$. Substitute $\|\mathrm{P}_\Omega(\mathbf{Z}_U\mathbf{V} - \tilde{\mathbf{A}}) + \cdot\mathrm{P}_\phi(\mathbf{Z}_U\mathbf{V} - \mathbf{U}\mathbf{V})\|_F^2$ with $\|\mathrm{P}_\Omega(\mathbf{Z}_U\mathbf{V} - \tilde{\mathbf{A}})\|_F^2 + \|\mathrm{P}_\phi(\mathbf{Z}_U\mathbf{V} - \mathbf{U}\mathbf{V})\|_F^2$ into $\mathrm{S}_U(\mathbf{U}|\mathbf{U}, \mathbf{V})$, we get

$$\mathrm{S}_U(\mathbf{Z}_U|\mathbf{U}, \mathbf{V}) - F(\mathbf{Z}_U, \mathbf{V}) = \|\mathrm{P}_\phi(\mathbf{Z}_U\mathbf{V} - \mathbf{U}\mathbf{V})\|_F^2 \geq 0 \tag{37}$$

In a similar way, we can immediately obtain

$$\mathrm{S}_V(\mathbf{Z}_V|\mathbf{U}, \mathbf{V}) - F(\mathbf{U}, \mathbf{Z}_V) = \|\mathrm{P}_\phi(\mathbf{U}\mathbf{Z}_V - \mathbf{U}\mathbf{V})\|_F^2 \geq 0 \tag{38}$$

Now we consider the update of $\mathbf{U}$,

$$\mathrm{S}_U(\tilde{\mathbf{U}}^{(l)}|\bar{\mathbf{U}}^{(l)}, \bar{\mathbf{V}}^{(l)}) = \min_{\mathbf{Z}_U} \mathrm{S}_U(\mathbf{Z}_U|\bar{\mathbf{U}}^{(l)}, \bar{\mathbf{V}}^{(l)}) \leq \mathrm{S}_U(\bar{\mathbf{U}}^{(l)}|\bar{\mathbf{U}}^{(l)}, \bar{\mathbf{V}}^{(l)}) \tag{39}$$

Using (33) and (34), we have

$$F(\tilde{\mathbf{U}}^{(l)}, \bar{\mathbf{V}}^{(l)}) \leq \mathrm{S}_U(\tilde{\mathbf{U}}^{(l)}|\bar{\mathbf{U}}^{(l)}, \bar{\mathbf{V}}^{(l)}) \tag{40}$$

$$\mathrm{S}_U(\bar{\mathbf{U}}^{(l)}|\bar{\mathbf{U}}^{(l)}, \bar{\mathbf{V}}^{(l)}) = F(\bar{\mathbf{U}}^{(l)}, \bar{\mathbf{V}}^{(l)}) \tag{41}$$

Combining these and (39) leads to the inequality

$$F(\tilde{\mathbf{U}}^{(l)}, \bar{\mathbf{V}}^{(l)}) \leq F(\bar{\mathbf{U}}^{(l)}, \bar{\mathbf{V}}^{(l)}) \tag{42}$$

Similarly, by the definition of $\tilde{\mathbf{V}}^{(l)}$,

$$\mathrm{S}_V(\tilde{\mathbf{V}}^{(l)}|\tilde{\mathbf{U}}^{(l)},\bar{\mathbf{V}}^{(l)}) = \underset{\mathbf{Z}_V}{min}\mathrm{S}_V(\mathbf{Z}_V|\tilde{\mathbf{U}}^{(l)},\bar{\mathbf{V}}^{(l)}) \leq \mathrm{S}_V(\bar{\mathbf{V}}^{(l)}|\tilde{\mathbf{U}}^{(l)},\bar{\mathbf{V}}^{(l)}) \tag{43}$$

Using (33), (34) and (43), we obtain

$$F(\tilde{\mathbf{U}}^{(l)},\tilde{\mathbf{V}}^{(l)}) \leq F(\tilde{\mathbf{U}}^{(l)},\bar{\mathbf{V}}^{(l)}) \tag{44}$$

By combining (42) and (44), we can derive the inequality

$$F(\tilde{\mathbf{U}}^{(l)},\tilde{\mathbf{V}}^{(l)}) \leq F(\tilde{\mathbf{U}}^{(l)},\bar{\mathbf{V}}^{(l)}) \leq F(\bar{\mathbf{U}}^{(l)},\bar{\mathbf{V}}^{(l)}) \tag{45}$$

This completes the proof. □

Theorem 4. *Assume that the row vectors (node representations) of $\mathbf{H}^{(0)}$ are drawn from a union of independent subspaces $\{S_i\}_{i=1}^{c}$. Also Assume that the update rule of node representation matrix is*

$$\mathbf{H}^{(l+1)} = (1-\beta)\mathbf{Z}^{(l)}\mathbf{H}^{(l)} + \beta\mathbf{H}^{(0)}, \tag{46}$$

where $\mathbf{Z}^{(l)}$ is an optimal solution (assume it exists) to the following optimization problem

$$min \quad \|\mathbf{Z}\|_* + \lambda\|\mathbf{Z}\|_F^2, \quad s.t. \quad \mathbf{H}^{(l)} = (1-\beta)\mathbf{Z}\mathbf{H}^{(l)} + \beta\mathbf{H}^{(0)}, \tag{47}$$

where $\lambda > 0$, $\beta > 0$. Then for any node pair v_i and v_j that belong to different subspaces, we always have $\mathbf{Z}_{i,j}^{(l)} = 0$, $\forall l \geq 0$.

Proof. Without loss of generality, we assume that $\mathbf{H}^{(0)}$ has been rearranged such that $\mathbf{H}^{(0)} = [\mathbf{H}_1^T, ..., \mathbf{H}_c^T]^T$, where nodes in $\mathbf{H}_i$ belong to the same subspace. We first consider the situation of $l = 0$. The problem becomes

$$\min \quad \|\mathbf{Z}\|_* + \lambda\|\mathbf{Z}\|_F^2, \quad s.t. \quad \mathbf{H}^{(0)} = (1-\beta)\mathbf{Z}\mathbf{H}^{(0)} + \beta\mathbf{H}^{(0)} \tag{48}$$

We prove this by contradiction. Assume there exists an optimal solution $\mathbf{Z}^*$ to (48) with at least one $\mathbf{Z}_{i,j}^* \neq 0$ while v_i and v_j belong to different subspaces. We define a matrix $\mathbf{W}$ as

$$\mathbf{W}_{i,j} = \begin{cases} \mathbf{Z}_{i,j}^* & \text{if node pair } (i,j) \text{ belong to the same subspace} \\ 0 & \text{otherwise} \end{cases} \tag{49}$$

Write $\mathbf{Q} = \mathbf{Z}^* - \mathbf{W}$, which satisfies

$$\mathbf{Q}_{i,j} = \begin{cases} 0 & \text{if node pair } (i,j) \text{ belong to the same subspace} \\ \mathbf{Z}_{i,j}^* & \text{otherwise} \end{cases} \tag{50}$$

Let v_j belong to the l-th subspace. By definition, $[\mathbf{W}\mathbf{H}^{(0)}]_{j,:} \in S_l$ and $[\mathbf{Q}\mathbf{H}^{(0)}]_{j,:} \in \oplus_{i\neq l}S_i$. Noticing that

$$[\mathbf{Q}\mathbf{H}^{(0)}]_{j,:} = [\mathbf{Z}^*\mathbf{H}^{(0)}]_{j,:} - [\mathbf{W}\mathbf{H}^{(0)}]_{j,:} = \mathbf{H}_{j,:}^{(0)} - [\mathbf{W}\mathbf{H}^{(0)}]_{j,:}, \tag{51}$$

we have $[\mathbf{Q}\mathbf{H}^{(0)}]_{j,:} \in S_l$. Since the subspaces are independent, we obtain $[\mathbf{Q}\mathbf{H}^{(0)}]_{j,:} = \mathbf{0}$. This leads to $\mathbf{W}\mathbf{H}^{(0)} = \mathbf{Z}^*\mathbf{H}^{(0)} - \mathbf{Q}\mathbf{H}^{(0)} = \mathbf{Z}^*\mathbf{H}^{(0)}$. Therefore,

$$\mathbf{H}^{(0)} = (1-\beta)\mathbf{W}\mathbf{H}^{(0)} + \beta\mathbf{H}^{(0)} \tag{52}$$

$\mathbf{W}$ is a feasible solution. Note that $\mathbf{H}^{(0)}$ has been properly rearranged. We can write

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & 0 & 0 & 0 \\ 0 & \mathbf{W}_2 & 0 & 0 \\ 0 & 0 & ... & 0 \\ 0 & 0 & 0 & \mathbf{W}_c \end{bmatrix}_{n\times n} \tag{53}$$

Using the well-known lemma (see, e.g., (Horn & Johnson, 1991), Theorem 3.4.1)

$$\left\| \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} \right\|_* \geq \left\| \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{pmatrix} \right\|_* = \|\mathbf{A}\|_* + \|\mathbf{D}\|_*, \tag{54}$$

we obtain $\|\mathbf{W}\|_* \leq \|\mathbf{Z}^*\|_*$. Noticing that there exists at least one $\mathbf{Z}^*_{i,j} \neq 0$ while v_i and v_j belong to different subspaces, we have $\|\mathbf{W}\|_F^2 < \|\mathbf{Z}^*\|_F^2$. Hence,

$$\|\mathbf{W}\|_* + \lambda\|\mathbf{W}\|_F^2 < \|\mathbf{Z}\|_* + \lambda\|\mathbf{Z}\|_F^2 \tag{55}$$

Combining this with (52), we can find a feasible solution $\mathbf{W}$ which has a smaller value of objective function than $\mathbf{Z}^*$. Therefore, $\mathbf{Z}^*$ does not optimize (48), which contradicts the assumption that $\mathbf{Z}^*$ is an optimal solution. We can conclude that all the optimal solutions to (48) have the desirable property that between-subspace elements are all zeros.

Now we consider

$$\mathbf{H}^{(1)} = (1-\beta)\mathbf{Z}^{(0)}\mathbf{H}^{(0)} + \beta\mathbf{H}^{(0)}, \tag{56}$$

where $\mathbf{Z}^{(0)}$ is an optimal solution to (48). Since the between-subspace elements are all zeros in $\mathbf{Z}^{(0)}$ and $\beta > 0$, the data matrix $\mathbf{H}^{(1)}$ also satisfies the assumption that the row vectors are drawn from a union of independent subspaces $\{S_i\}_{i=1}^c$ (the mapping from row vectors to subspaces remains the same). In a similar way, we can further prove that $\mathbf{Z}^{(1)}$ also satisfies $\mathbf{Z}^{(1)}_{i,j} = 0$ if (i,j) belong to different subspaces. The only difference lies in (51), which becomes,

$$[\mathbf{Q}\mathbf{H}^{(1)}]_{j,:} = [\mathbf{Z}^*\mathbf{H}^{(1)}]_{j,:} - [\mathbf{W}\mathbf{H}^{(1)}]_{j,:} = \frac{1}{1-\beta}(\mathbf{H}^{(1)}_{j,:} - \beta\mathbf{H}^{(0)}_{j,:}) - [\mathbf{W}\mathbf{H}^{(1)}]_{j,:} \tag{57}$$

Since $\mathbf{H}^{(1)}_{j,:} \in S_l$, we have $(\mathbf{H}^{(1)}_{j,:} - \beta\mathbf{H}^{(0)}_{j,:}) \in S_l$. Then $[\mathbf{Q}\mathbf{H}^{(1)}]_{j,:} \in S_l$ still holds. The remaining part can be proved in the same way.

Analogously, we can prove that $\mathbf{Z}^{(2)}$, $\mathbf{Z}^{(3)}$, ... all have this property. □

Remark 2. *Note that we can remove the nuclear norm term and use only $\|\mathbf{Z}\|_F^2$ as the objective function. It is easy to prove that the between-subspace elements of $\mathbf{Z}^{(l)}$ obtained are also zeros. However, this is prone to make the within-subspace elements sparse, which contradicts the goal to represent membership by $\mathbf{Z}^{(l)}$, namely $\mathbf{Z}^{(l)}_{i,j} \neq 0$ represents (i,j) belong to the same subspace while $\mathbf{Z}^{(l)}_{i,j} = 0$ represents (i,j) belong to different subspaces. In practice, using nuclear norm alone is sufficient to obtain an optimal solution with almost all the between-subspace elements being zeros. We include the term $\lambda\|\mathbf{Z}\|_F^2$ for theoretical completeness.*

A.9 Additional experimental results

Robustness to random Gaussian noise. Since the performance of LRGNN is affected by the generated signed adjacency matrix, we here examine the robustness of LRGNN to random noise added to the generated signed adjacency matrix. We consider a scenario where the signed adjacency matrix is corrupted by random Gaussian noise. Mathematically, we construct a corrupted signed adjacency matrix $\mathbf{A}_{noise} = \tilde{\mathbf{A}} + \mathbf{N}$, where $\mathbf{N}_{i,j} = \epsilon_{i,j}$, if $(i,j) \in \mathcal{E}$ and 0 otherwise, with $\epsilon_{i,j}$ i.i.d. sampled from a Gaussian distribution $\mathcal{N}(0,\sigma^2)$. Then this corrupted adjacency matrix is used for LRMF. We choose σ from $\{0.1, 0.2, ..., 1.0\}$. Figure 7 reports the decline of mean test accuracy caused by noise w.r.t. different standard deviations of Gaussian distributions. The declines are smaller than 3% for all the datasets. Note that the values of entries of $\tilde{\mathbf{A}}$ are within the range of $[-1,1]$. Hence, the noise may cause a considerable change in value, *i.e.*, $\frac{|\epsilon_{i,j}|}{|\tilde{\mathbf{A}}_{i,j}|} > 1$. The results imply that LRGNN is not sensitive to the quality of the generated signed adjacency matrix, which may be attributed to the LRR term of the objective function.

We point out two potentially feasible measures for promoting robustness. We can design a more sophisticated element-wise function $\mathrm{P}_\Omega(\cdot)$ that better reflects the importance of each observed entry, especially when the observed entries are noisy. We may define $\mathrm{P}_\Omega(\tilde{\mathbf{A}}_{i,j}) = W_{i,j}\tilde{\mathbf{A}}$, $\forall(i,j) \in \mathcal{E}$, where $W_{i,j}$ is the importance of the observed entry $\tilde{\mathbf{A}}_{i,j}$. The update rules of $\mathbf{U}$ and $\mathbf{V}$ are still correct once we ensure the definition of $\mathrm{P}_\phi(\cdot)$ satisfies $\mathrm{P}_\phi(\tilde{\mathbf{A}}) + \mathrm{P}_\Omega(\tilde{\mathbf{A}}) = \tilde{\mathbf{A}}$. Reputation is an important notion in social network based recommender systems (Tang et al., 2013), which captures the credibility of a user's ratings. The quality of the signed adjacency matrix $\tilde{\mathbf{A}}$ is affected by the pseudo labels. However, the pseudo labels generated by neural networks may be false. Let $\hat{\mathbf{Y}}_{i,:}$ denotes the probability distribution of node v_i. Intuitively, a uniform probability distribution implies

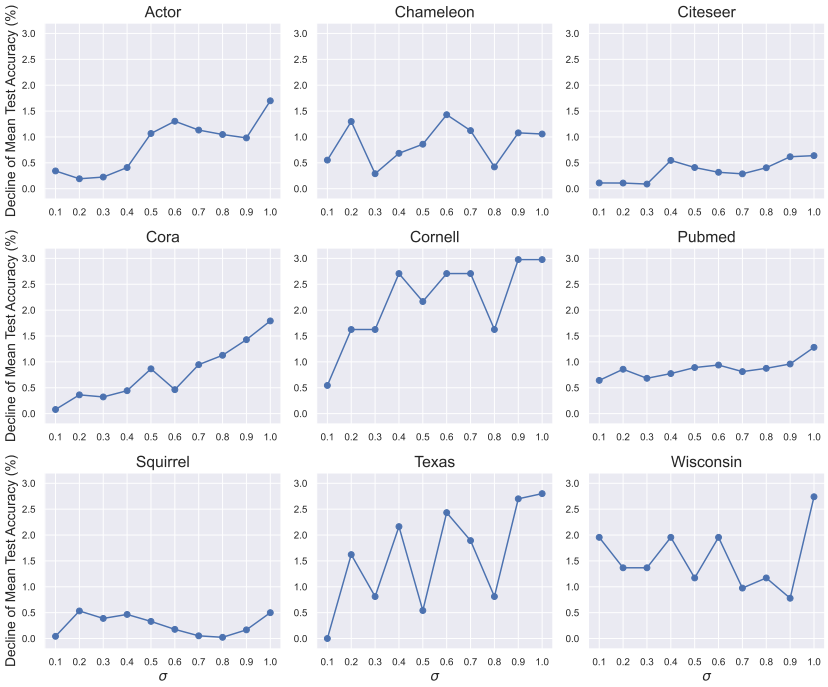

Figure 7: Decline of mean test accuracy w.r.t. different standard deviations of Gaussian noises.

	Texas	Wiscon.	Cornell	Actor	Squirrel	Chamel.	Cora	Citeseer	Pubmed
LRGNN	**89.19±4.49**	**88.23±3.04**	**86.22±6.10**	**37.10±2.12**	**74.51±1.90**	**78.93±1.23**	**88.23±1.03**	**77.46±1.31**	**89.60±0.54**
LRGNN-MF	86.20±4.90	84.71±4.75	81.35±6.67	35.69±0.95	58.39±4.38	64.78±4.06	87.88±1.03	77.30±1.39	89.05±0.43
LRGNN-Reg	88.38±2.43	82.94±4.11	83.51±5.85	36.61±1.17	65.83±2.19	69.12±0.84	86.92±0.96	75.57±1.60	87.36±0.25
LRGNN-Uni	87.84±3.67	82.94±4.72	85.14±7.66	34.53±1.08	69.45±1.78	68.46±1.38	87.67±1.38	77.29±1.29	89.39±0.18
LRGNN-DA	**89.19±3.67**	86.86±4.72	85.95±7.66	36.86±1.08	72.52±1.78	75.65±1.38	88.23±1.38	77.32±1.29	89.45±0.18

Table 4: Ablation study on propagation term.

that neural networks have no confidence that to which class the node belongs. Therefore, we can say that node v_i has a poor reputation if $\hat{\mathbf{Y}}_{i,:}$ is uniformly distributed and the importance of its rating should be reduced. Noticing that the Euclidean norm of $\hat{\mathbf{Y}}_{i,:}$ can measure its uniformity, we define the reputation of node v_i as $r_i = \|\hat{\mathbf{Y}}_{i,:}\|_2^2$. Then the importance of the observed entry $\tilde{\mathbf{A}}_{i,j}$ can be described by $W_{i,j} = r_i r_j$. This importance is well-defined since $0 < W_{i,j} \leq 1$.

Besides, we may explicitly model a noise matrix for matrix factorization. For example, we can use the following objective function.

$$F(\mathbf{U}^{(l)}, \mathbf{V}^{(l)}) = \|\mathbf{H}^{(l)} - (1-\beta)\mathbf{U}^{(l)}{\mathbf{V}^{(l)}}^T\mathbf{H}^{(l)} - \beta\mathbf{H}^{(0)}\|_F^2 + \gamma\|\mathrm{P}_\Omega(\mathbf{U}^{(l)}{\mathbf{V}^{(l)}}^T - \tilde{\mathbf{A}} - \mathbf{N})\|_F^2, \quad (58)$$

where $\mathbf{N}$ is a noise matrix, which can be modeled as mixtures of Gaussian and estimated by maximum likelihood estimation. We refer the interested reader to Meng & De La Torre (2013) for an instance.

Ablation study. We investigate the effectiveness of the components by conducting an ablation study. We consider four variants: LRGNN-MF only contains the MF term in the objective function; LRGNN-Uni indicates that the signed adjacency matrix is replaced with the uniform sparse adjacency matrix in GCN; LRGNN-Reg indicates that the matrix factorization term is replaced with a Frobenius term; LRGNN-DA drops the MLP(A) from $\mathbf{H}^{(0)}$. Since LRMF is the core of our method, we will not try to remove the MF item. We compare these variants with LRGNN and report the node classification results in Table 4. The results demonstrate that propagation term can consistently im-

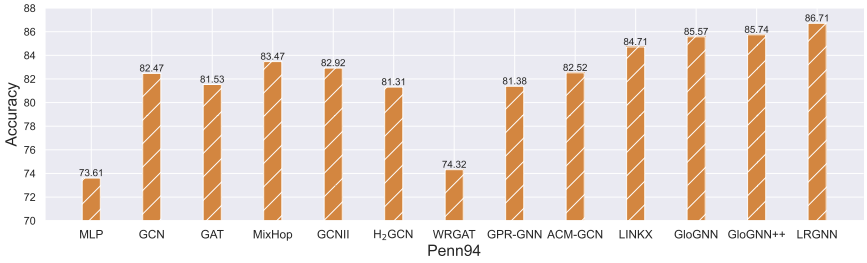

Figure 8: Result on Penn94 dataset.

prove the classification accuracy, especially for heterophilious datasets. Also, the uses of both MF and signs are crucial, dropping these components causes significant degradations in performance.

Aggregation coefficient study. We visualize the learned aggregation coefficients extracted from the last layer. The edges are divided into two categories based on whether the connected two nodes belong to the same class. To form a comparison, we also visualize the aggregation coefficients learned by GloGNN++ and FAGCN. Note that we only plot the learned weights of observed edges because, in FAGCN, nodes aggregate only their immediate neighbors during feature propagation. We can see from Figure 9 that for FAGCN and GloGNN++, the aggregation coefficient distribution of the intra-edges shows a similar pattern to that of the inter-edges, which implies that they cannot assign proper signs according to the label relationship. For LRGNN, there is a clear difference between the two distributions. It is worth noting that FAGCN uses the static attention function adapted from standard GAT layer, thus the ranking of the aggregation coefficients is unconditioned on the query node. For GloGNN++, it uses the term $\|\mathbf{Z} - \mathbf{A}\|_F^2$ where $\mathbf{A}$ only contains positive edge weights. In conclusion, LRGNN performs better in capturing the label agreement between nodes. This further shows the benefit of using the LRMF.

Results on a large-scale graph. Lim et al. (2021) proposed 7 large-scale non-homophilious datasets that allow comprehensive evaluation of GNNs in non-homophilous settings. However, these datasets are too large to run on our machines. Therefore, we cannot evaluate LRGNN on these datasets. We use the smallest dataset Penn94 for an experiment, the only dataset that we can run on our machines. We report the node classification result in Figure A.9. The result shows that LRGNN achieves state-of-the-art performance, which implies that LRGNN performs well on large-scale graphs

A.10 Experimental setup

We implement LRGNN with Pytorch Geometry Library. We ran our experiments on an Nvidia P100 GPU with 16GB of memory. For real-world graphs, we use 10 random splits ($48\%, 32\%20\%$ for training/validation/testing) provided by Pei et al. (2020) and available from Fey & Lenssen (2019). For Table 1, we directly use the available results from Yan et al. (2021); Lim et al. (2021). For the results on synthetic graphs, we run the baseline methods using the codes released by their authors and fine tune hyper-parameters based on validation set. We perform a grid search to tune hyper-parameters based on the validation set, as shown in Table 5.

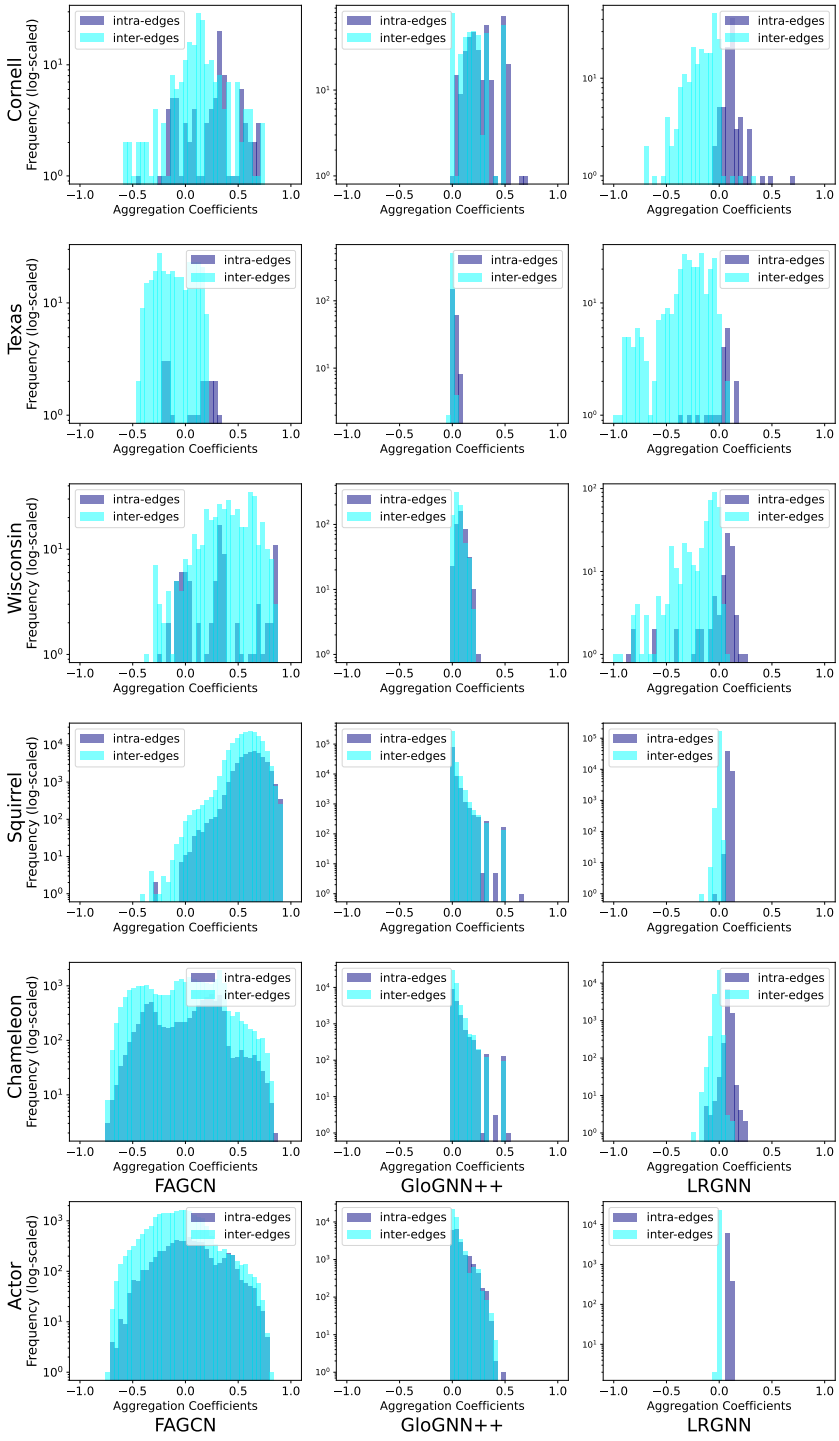

Figure 9: Visualization results of aggregation coefficients learned by different methods on heterophilious datasets. For LRGNN, some aggregation coefficients are not in the range of [-1,1].

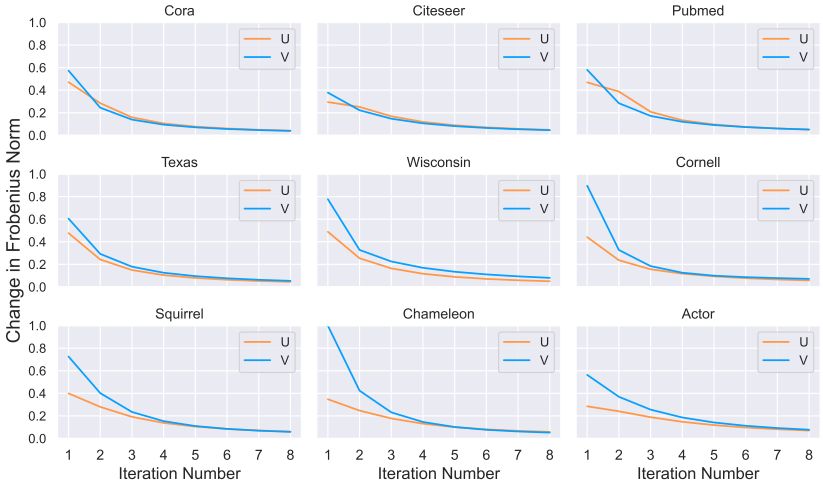

Figure 10: Convergence rate of the softImpute-ALS. Both $\mathbf{U}$ and $\mathbf{V}$ converge within 8 iterations. Here, we specify $l = L$ (the last layer). Change is calculated as $\frac{\|\mathbf{U}^k - \mathbf{U}^{k-1}\|_F^2}{\|\mathbf{U}^k\|_F^2}$, where k denotes the k-th iteration.

Hyper-parameter	**Range**
learning rate	$\{0.01, 0.005\}$
weight decay	$\{5e-3, 5e-4\}$
dropout	$[0, 0.9]$
early stopping	$\{40, 100, 200\}$
β	$[0, 0.9]$
μ	$[0, 0.9]$
δ	$[0, 0.9]$
γ	$\{50, 100, 200, 1000, 1500\}$
number of layers	$\{1\}$
number of iterations	$\{1, 2\}$
Estimator to generate pseudo labels	$\{\text{GCN}, \text{MLP}\}$

Table 5: Search space for hyper-parameters.