Cross-links Matter for Link Prediction: Rethinking the Debiased GNN from a Data Perspective (Supplemental Materials)

Anonymous Author(s)
Affiliation
Address
email

A Training Algorithm

Algorithm 1 Proposed training process

Input: Graph $\mathcal{G}$ with link set $\mathcal{E}^O$. Hyper-parameters: α, β, and T, learning rate γ^O, γ^A.
Output: Node embeddings $\mathbf{Z}$

```
Randomly initialize twins GNN models with θ^O, θ^A, embedding fusion module with θ^F.
Split G into |C| communities and categorize links into internal-links and cross-links.
Select augmented supervision signals E^A with the highest Jaccard coefficient or co-occurrence frequency.
while not converged do
    Compute L^O and L^A by Eq.(4)
    Update twins GNN models: θ^O ← θ^O + γ^O · ∇_{θ^O} L^O,    θ^A ← θ^A + γ^A · ∇_{θ^A} L^A
    Compute learning rate γ_t^F and step size S_t by Eq.(8)
    for step = 1 to S_t do
        Compute L^F by Eq.(7)
        Update embedding fusion module: θ^F ← θ^F + γ_t^F · ∇_{θ^F} L^F
        Update GNN models: θ^O ← θ^O + γ_t^F · ∇_{θ^O} L^F,    θ^A ← θ^A + γ_t^F · ∇_{θ^A} L^F
    end for
end while
return Z
```

Here we provide the pseudo codes of our training process, which are the core components helping GNNs to address the bias between internal-links and cross-links without compromising utility. The algorithm is also literally described in Section 3.6 for a better understanding.

B Further Analysis on the Role of Cross-links

B.1 The relationship between cross-links and information cocoons

To fully understand the relationship between cross-links and information cocoons, we conduct the following experiments for analysis.

• **Experimental settings.** Based on the communities detected by Louvain algorithm [1] in advance, we get the internal-links and cross-links of a network, and here we take Epinions and DBLP, two real-world social networks as examples. The detailed dataset information is described in Section 4.1. Next, we borrow the concept of message propagation from the Friedkin-Johnsen dynamics model

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

[5] and revise its formula to simulate the information propagation with randomly initialized node embeddings:

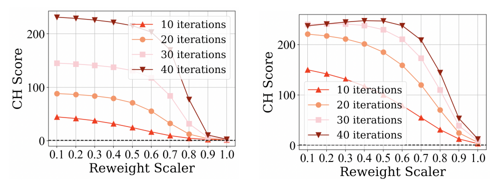
$$\mathbf{Z}_i^t = \frac{\mathbf{Z}_i^{t-1} + \sum_{j \in \mathcal{N}_i} w_{ij} \mathbf{Z}_j^{t-1}}{|\mathcal{N}_i| + 1} \tag{1}$$

where $\mathbf{Z}_i^t$ denotes the embedding of node i at iteration t, and $\mathcal{N}_i$ represent the neighbors of node i. w_{ij} is a manually controllable reweight scaler determined by the type of link $\langle i, j \rangle$. At each iteration, each node will update its embedding with the weighted average embeddings from its neighbors and itself. For simulating the lack of cross-links, we weaken the role of cross-links in information propagation by tuning the w for cross-links from 1 to 0, and setting w for internal-links to 1.

We further use Calinski-Harabasz(CH) index [4, 6] to measure the extent of the information cocoons phenomenon in a network, which can be calculated as follows:

$$CH_C = \frac{SSB_C(\mathbf{Z}^t)}{SSW_C(\mathbf{Z}^t)} \cdot \frac{N - C}{C - 1} \tag{2}$$

where SSW_C and SSB_C are functions to measure the within-cluster dispersion and between-cluster dispersion, respectively [6]. N denotes the number of nodes, and C denotes the number of communities. A higher CH index score indicates that node embeddings are more polarized among communities, which further illustrates the extent of the information cocoon problem in the current network.



(a) Epinions (b) DBLP

Figure C1: The distribution of CH index score wrt. the weight of cross-links during propagation. A higher value indicates more severe information cocoons. The dashed lines indicate the CH index score under a normal setting.

• **Experimental results and analysis.** In Figure C1 we show the CH index scores with node embeddings at different propagation iteration t, and we can observe that, as the information propagation weight w for cross-links decreases, the CH index score increases consistently and far exceeds that in normal settings ($w = 1$), which indicates that the final node embeddings present more serious polarization problems among communities. Since the information in a single community is relatively limited as shown in Figure 2, the information cocoon problem actually becomes more severe with the lack of cross-links.

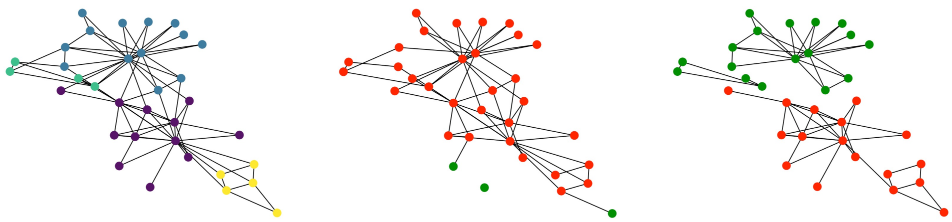
B.2 The relationship between cross-links and graph conectivity

With borrowing the concept of network diffusion, we try to explore the role of cross-links in graph conectivity in this part. Specifically, we apply a classic model in network diffusion: the SI model [11], to simulate the process of information propagation. In this model, each node is randomly initialized with a status called *susceptible* or *infected* at the beginning. During the diffusion iteration process, SI assumes that each infected node could infect its susceptible neighbors with probability p, and once a node becomes infected, it stays infected until the end of network diffusion, i.e. there are no more new infected nodes in a new iteration.

In order to provide a clear and vivid illustration, we take one of the most representative social networks – Zachary's karate club[1] as an example. All nodes are divided into four non-overlapping communities by Louvain algorithm [1] in advance, and node #0, which stands for the instructor in this club, is initialized as the only infected node at iteration 0. We further randomly remove 80% cross-links in the graph before starting the simulation. For getting a more convincing conclusion, the network diffusion process on a graph with the same number of random edges removed is also simulated for comparison.

The final visualization results are shown in Figure C2. It can be seen that although we remove some edges randomly from the whole graph, the infected node #0 still propagates its information to almost all nodes in the club (red nodes in the figure) successfully. In contrast, after removing the same number of cross-links, there appears to be an obvious information isolation phenomenon, and nearly half of the nodes remain *susceptible*. In other words, cross-links play a role in bridging two different communities during network diffusion, and it would be hard for a node to send or receive messages

[1] https://en.wikipedia.org/wiki/Zachary%27s_karate_club



(a) Original graph (b) Drop links randomly (c) Drop cross-links

Figure C2: The visualization of diffusion simulation with SI model. Red nodes denote *infected* nodes, and green nodes represent *susceptible* nodes. (a) The original graph of the karate club. Nodes with the same color denote a community. (b) Infected graph after dropping some random edges. (c) Infected graphs after dropping some cross-links.

from other communities without enough cross-links. In this way, the existence of cross-links plays a key role in preserving graph connectivity.

C Additional Experimetal Settings

C.1 Baselines

Here we introduce additional details for the base models and baselines used in our experiments.

(1) Base models.

- **GraphSAGE** [7]: GraphSAGE is an inductive learning framework for generating node embeddings, which samples a fixed number of neighbors during aggregation to alleviate the "neighborhood explosion" issues.
- **GIN** [20]: GIN is a graph neural network that is theoretically as powerful as the Weisfeiler-Lehman test with injective aggregation, combination, and readout functions.
- **GAT** [18]: GAT deploys the attention mechanism during aggregation to capture the neighborhood information with different weights.
- **PPRGo**[2]: By utilizing a Personalized Page Rank matrix to approximate the propagation and aggregation steps with multi-layer graph convolution, PPRGo greatly improves the efficiency and effectiveness on large graphs.
- **LightGCN** [8]: LightGCN empirically finds the redundancy of feature transformation and non-linear activation functions, and greatly simplifies the model architecture with even higher performance on the recommendation tasks.
- **UltraGCN** [14]: Based on LightGCN, UltraGCN further theoretically simplifies the model architecture with approximating the infinite-layer information propagation and aggregation.

(2) Baselines.

- **FairWalk** [16]: Instead of random walk in node2vec, FairWalk chooses its next hop by considering the sensitive attributes in the neighborhood, which successfully mitigates the unfairness related to the sensitive attribute.
- **CFC** [3]: To ensure that the learned embeddings are not correlated with sensitive attributes, such as age or gender, CFC introduces an adversarial framework to enforce fairness constraints.
- **FairAdj** [12]: With learning to assign each edge with different weights, FairAdj generates a fair adjacency matrix and greatly improves the dyadic fairness with comparable utility in link prediction tasks.
- **FLIP** [15]: Concentrating on bursting the filter bubbles in social networks from a dyadic fairness perspective, FLIP also utilizes an adversarial learning framework to generate non-sensitive node embeddings for further link prediction.

- **UGE** [19]: UGE aims at learning unbiased graph embeddings from an unobserved graph, which involves no sensitive information, and further derives three kinds of variants namely UGE-w, UGE-r and UGE-c.

C.2 Evaluation metirc.

In this work, we apply Hits@50, which is widely adopted in other researches [21, 22] and OGB leaderboard [9], as our main evaluation metric to measure the link prediction performance of different GNN models. The Hits@50 can be computed by:

$$Hits@50 = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \mathbb{I}(rank_i < 50) \tag{3}$$

where N_{test} represents the sample size of test set, and $\mathbb{I}$ represents an indicator function. $rank_i$ denotes the similarity ranking of the ith sample.

C.3 Reproducibility

Dataset. For each dataset, we randomly sample and remove 5% of links in the original graph to construct the validation set and test set, and the remaining links are treated as the training set. Each true sample will be ranked among a set of 100000 randomly sampled negative links for evaluation[2]. Note that, there is no side information, such as node features or link attributes, involved during our experiments, and we assign each node on the graph with a learnable embedding vector for training.

Hyper-parameters. As a model-agnostic framework, we deploy six kinds of GNN models as backbones, including GraphSAGE [7], GIN [20], GAT [18], PPRGo [2], LightGCN [8] and UltraGCN [14]. For all these models, we set the output embedding dimension as 64. The layer of the embedding fusion module is set to 1. The learning rates for twin GNNs are both set as 0.001 after grid search. As for the hyper-parameters in Eq.(8), α is set to be 0.005, and T is selected from {10, 25, 50} depending on the datasets and base models, and β is set to be 20. Augmentation ratio k is searched from {0.75, 1.0, 1.25} for each dataset. Both weight decay and dropout rate are set to 0.

In particular, for GraphSAGE, we adopt a mean-pooling during aggregation; for GIN, we apply a linear layer to update node features and use max-pooling during aggregation; for GAT, we use 4 attention heads in each layer; for PPRGo, we set α as 0.3, the walk length as 100; for LightGCN, we set the layer number as 2 and use the final layer's output as embeddings; for UltraGCN, we set the number of negative samples as 64, λ as 0.8, γ as 3.5. Our implementation code and datasets are released anonymously in `https://anonymous.4open.science/r/Neurips2023_9342/`.

For Fairwalk, we follow the settings in the original paper and set the walk number to 20, and the window length to 80; for CFC, we set the training steps of the discriminator as 5; for FairAdj, we set T_2 to 15 and λ to 10; for FLIP, we take the suggestions in the original implementation, and the settings are α(0.1), β(0.2); for UGE, we deploy the weighting-based variant as our baseline given that there is no non-sensitive attribute in our settings.

C.4 Details on LastFM dataset

Due to the heterogeneity of the recommendation datasets, it's hard to directly deploy community detection algorithms on the original networks and define the corresponding internal-links and cross-links.To this end, inspired by ItemCF [13, 17], we first generate an item-item graph according to the co-occurrence relationship. For example, given a pair of items $< v_1, v_2 >$, if they both have interactions with user u at least O times, where O is a hyperparameter to control the confidence of generated item-item graph, there will form a link between v_1 and v_2. Next,

Table C1: The comparison between our implementations and normal implementations on LastFM. The average results are reported after repeating each method five times.

		LastFM (Hits@50)
LightGCN	Original	$29.82\% \pm 0.25$
	Ours	$29.30\% \pm 0.21$
UltraGCN	Original	$28.50\% \pm 0.28$
	Ours	$27.32\% \pm 0.19$

[2] Here we follow the evaluation protocol in OGB [9], which is widely used in research.

we can deploy our framework on the generated item-item graph for learning debiased item embeddings $\mathbf{I} \in \mathbb{R}^{|V| \times D}$, where $|V|$ represents the number of items and D represents the embedding dimension. After that, item similarity matrix $\mathbf{I}^2 \in \mathbb{R}^{|V| \times |V|}$ is calculated, which is further used for the final recommendation:

$$\mathbf{P} = \mathbf{A} \times \mathbf{I}^2 \tag{4}$$

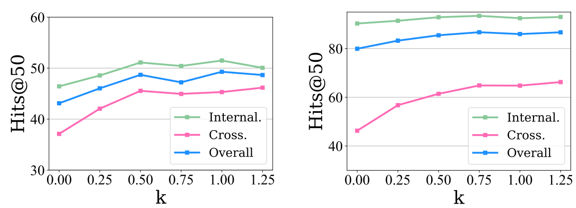
where $\mathbf{P} \in \mathbb{R}^{|U| \times |V|}$ denotes the predicted confidence matrix between users and items, and $\mathbf{A} \in \mathbb{R}^{|U| \times |V|}$ represents the adjacency matrix of the original user-item graph.

In order to prove that our implementation will not affect the performance of the original GNN models, we compared our implementation (denoted as "Ours"), where O is set to be 1, with GNNs trained on the user-item graph normally (denoted as "Original") in Table C1. The results indicate that our implementation does not sacrifice the capability of base GNN models severely to adapt our framework.

D Further Experiments and Analysis

D.1 Hyper-parameter Analysis

As the core component in our framework, supervision augmentation plays a key role in mitigating the bias between internal-links and cross-links. To explore its impact in a finer granularity, we vary the augmentation ratio k and see how the performance of our method changes. Specifically, we take LightGCN as the base model and investigate the performance on internal-links (Internal.), cross-links (Cross.) and the whole link set (Overall) by varying k in {0, 0.25, 0.5, 0.75, 1, 1.25}. Without losing generality, here we take Jaccard based supervision augmentation. As shown in Figure C3, the performance of the two kinds of links increasingly improves as k grows, accompanied by a steady decrease in the difference between them. This is expected because we introduce a large amount of augmented cross-links signals to mitigate the bias. And when k reaches 1, which means $|\mathcal{E}_{in}| \approx |\mathcal{E}_{cr}|$, the framework gradually converges to a stable status. Empirically, a setting $k = 1$ would be a near-optimal option.



(a) Epinions (b) DBLP

Figure C3: The impact of augmentation size

D.2 Alternative Community Detection Algorithm

In this part, we aim to conduct an ablation study on different community detection methods to prove the usefulness of our proposed framework. Since we emphasize the bias from a topological perspective, we prefer to use the community detection algorithm based on graph structure. Specifically, to illustrate the universality of our framework, we conduct experiments based on the METIS [10] algorithm as an ablation study, and the results on three GNNs are shown in Table C2. Specifically, the number of communities is set to be 50 in advance for METIS, and all other hyper-parameters are set to be the same as that in Louvain-based experiments, which can be found in Section B.3. All results are based on Jaccard based augmentation.

Table C2: Ablation study with METIS community detection algorithm on two real-world datasets. The results (Hits@50) are reported in percentage (%). We **bold** the results when our framework improves the base GNN model.

		Epinions				DBLP			
		Internal.↑	Cross.↑	Overall↑	Bias↓	Internal.↑	Cross.↑	Overall↑	Bias↓
SAGE	Orig.	36.97	19.48	30.75	17.49	69.80	19.00	54.28	50.80
	Debias	**39.06**	**28.93**	**35.89**	**10.13**	**78.67**	**34.65**	**65.22**	**44.02**
GAT	Orig.	38.33	34.77	37.15	3.56	68.62	28.15	56.26	40.47
	Debias	**39.96**	**36.88**	**38.86**	**3.08**	**75.94**	**42.77**	**65.81**	**33.17**
UlltraGCN	Orig.	27.27	11.62	20.77	15.65	97.34	70.57	89.16	26.77
	Debias	**46.92**	**38.18**	**44.04**	**8.74**	**97.47**	**73.67**	**90.20**	**23.80**

The results indicate that, although we change the community detection algorithm, our framework still successfully mitigates the bias between internal-links and cross-links, and achieves competitive results compared with the Louvain-based results, which verifies our work's compatibility.

D.3 Supervision Augmentation Analysis

In Section 3.3, we design two kinds of data augmentation methods for generating pseudo cross-links supervision signals. Intuitively, if the pseudo supervision signals have a high confidence level, they can provide significant benefits to our framework. To this end, we aim to verify our hypothesis and analyze the impact of different supervision augmentation methods on our framework.

We first statistic the average hop distance between the node pairs generated with different supervision augmentation methods. As shown in Table C3, since we only choose node pairs with the most common neighbors in Jaccard based augmentation, the hop distance is fixed to 2. When we use random walk based augmentation, the average distance increases consistently on two datasets, which verifies its effectiveness in covering nodes that are not located in the boundary of communities.

Table C3: The average hop distance between node pairs generated by different supervision augmentation methods.

	Epinions	**DBLP**
Jaccard based	2.00	2.00
Random walk based	2.69	3.14

Table C4: Link prediction performance (Hits@50) of internal-links, cross-links and the whole link set of our methods with random walk based augmentation and corresponding base models on two real-world datasets. The results are reported in percentage (%). We **bold** the results when our framework improves the base GNN model.

		Epinions				**DBLP**			
		Internal.↑	Cross.↑	Overall↑	Bias↓	Internal.↑	Cross.↑	Overall↑	Bias↓
SAGE	Orig.	31.68	28.91	30.69	2.77	69.27	14.62	56.41	54.65
	Fair.	**31.72**	**29.17**	**31.28**	**2.55**	**80.28**	**28.63**	**68.12**	**51.65**
GIN	Orig.	33.49	30.97	32.59	2.52	56.66	16.86	47.29	39.80
	Fair.	**38.35**	**36.89**	**37.12**	**1.46**	**68.12**	**32.07**	**59.64**	**36.05**
GAT	Orig.	39.30	34.90	37.73	4.40	66.25	22.47	55.94	43.78
	Fair.	**40.02**	**36.29**	**37.98**	**3.73**	**75.03**	**32.18**	**64.94**	**42.85**
PPRGo	Orig.	42.86	28.75	37.83	14.11	85.71	41.14	75.28	44.58
	Fair.	**45.36**	**40.12**	**43.49**	**5.24**	**90.48**	**49.51**	**80.47**	**42.08**
LightGCN	Orig.	46.43	37.11	43.11	9.32	85.95	47.41	76.88	38.54
	Fair.	**48.15**	**40.45**	**45.41**	**7.65**	**92.16**	**57.55**	**84.01**	**34.61**
UlltraGCN	Orig.	30.62	5.81	21.78	24.81	95.74	63.82	88.22	31.92
	Fair.	**52.16**	**51.99**	**52.10**	**0.17**	**96.47**	**66.25**	**89.35**	**30.22**

Table C4 further presents the performance of our framework with random walk based augmentation, which is literally described in Section 3.3. Specifically, for providing fair comparison, all hyper-parameters in random walk based experiments, including augmentation ratio k and others are set to be the same as that in Jaccard based experiments. Compared to Table 2, it can be shown that the random walk based framework shows less improvement on cross-links, which results in worse debias results. This observation can be explained by the hop distance in Table C3, which implies that the random walk based augmentation may have lower confidence due to the longer topological distance between node pairs. However, compared with the base GNNs, the random walk based framework can still consistently reduce the bias between internal-links and cross-links with improved overall performance.

References

[1] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[2] A. Bojchevski, J. Klicpera, B. Perozzi, A. Kapoor, M. Blais, B. Rózemberczki, M. Lukasik, and S. Günnemann. Scaling graph neural networks with approximate pagerank. In R. Gupta, Y. Liu, J. Tang, and B. A. Prakash, editors, *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 2464–2473. ACM, 2020.

[3] A. J. Bose and W. L. Hamilton. Compositional fairness constraints for graph embeddings. In K. Chaudhuri and R. Salakhutdinov, editors, *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*, pages 715–724. PMLR, 2019.

[4] T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[5] U. Chitra and C. Musco. Analyzing the impact of filter bubbles on social network polarization. In J. Caverlee, X. B. Hu, M. Lalmas, and W. Wang, editors, *WSDM '20: The Thirteenth ACM*

International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020, pages 115–123. ACM, 2020.

[6] Y. Ge, S. Zhao, H. Zhou, C. Pei, F. Sun, W. Ou, and Y. Zhang. Understanding echo chambers in e-commerce recommender systems. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 2261–2270. ACM, 2020.

[7] W. L. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1024–1034, 2017.

[8] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 639–648. ACM, 2020.

[9] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[10] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, 1998.

[11] W. O. Kermack and A. G. McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, 115(772):700–721, 1927.

[12] P. Li, Y. Wang, H. Zhao, P. Hong, and H. Liu. On dyadic fairness: Exploring and mitigating bias in graph connections. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[13] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Comput.*, 7(1):76–80, 2003.

[14] K. Mao, J. Zhu, X. Xiao, B. Lu, Z. Wang, and X. He. Ultragcn: Ultra simplification of graph convolutional networks for recommendation. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 1253–1262. ACM, 2021.

[15] F. Masrour, T. Wilson, H. Yan, P. Tan, and A. Esfahanian. Bursting the filter bubble: Fairness-aware network link prediction. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 841–848. AAAI Press, 2020.

[16] T. Rahman, B. Surma, M. Backes, and Y. Zhang. Fairwalk: Towards fair graph embedding. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3289–3295. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[17] Y. Shen, Y. Wu, Y. Zhang, C. Shan, J. Zhang, K. B. Letaief, and D. Li. How powerful is graph convolution for recommendation? In G. Demartini, G. Zuccon, J. S. Culpepper, Z. Huang, and H. Tong, editors, *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 1619–1629. ACM, 2021.

[18] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. 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.

[19] N. Wang, L. Lin, J. Li, and H. Wang. Unbiased graph embedding with biased graph observations. In F. Laforest, R. Troncy, E. Simperl, D. Agarwal, A. Gionis, I. Herman, and L. Médini, editors, *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 1423–1433. ACM, 2022.

[20] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[21] S. Yun, S. Kim, J. Lee, J. Kang, and H. J. Kim. Neo-gnns: Neighborhood overlap-aware graph neural networks for link prediction. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 13683–13694, 2021.

[22] M. Zhang, P. Li, Y. Xia, K. Wang, and L. Jin. Labeling trick: A theory of using graph neural networks for multi-node representation learning. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 9061–9073, 2021.