

# Cross-links Matter for Link Prediction: Rethinking the Debiased GNN from a Data Perspective

Zihan Luo<sup>1</sup>, Hong Huang<sup>1\*</sup>, Jianxun Lian<sup>2</sup>, Xiran Song<sup>1</sup>, Xing Xie<sup>2</sup>, Hai Jin<sup>1</sup>

<sup>1</sup> National Engineering Research Center for Big Data Technology and System,  
Service Computing Technology and Systems Laboratory,  
Cluster and Grid Computing Lab,

School of Computer Science and Technology,  
Huazhong University of Science and Technology, Wuhan, China

<sup>2</sup> Microsoft Research Asia, Beijing, China  
{zihanluo, honghuang, xiransong, hjin}@hust.edu.cn,  
{jianxun.lian, xingx}@microsoft.com

## Abstract

Recently, the bias-related issues in GNN-based link prediction have raised widely spread concerns. In this paper, we emphasize the bias on links across different node clusters, which we call cross-links, after considering its significance in both easing information cocoons and preserving graph connectivity. Instead of following the objective-oriented mechanism in prior works with compromised utility, we empirically find that existing GNN models face severe data bias between internal-links (links within the same cluster) and cross-links, and this inspires us to rethink the bias issue on cross-links from a data perspective. Specifically, we design a simple yet effective twin-structure framework, which can be easily applied to most GNNs to mitigate the bias as well as boost their utility in an end-to-end manner. The basic idea is to generate debiased node embeddings as demonstrations and fuse them into the embeddings of original GNNs. In particular, we learn debiased node embeddings with the help of augmented supervision signals, and a novel dynamic training strategy is designed to effectively fuse debiased node embeddings with the original node embeddings. Experiments on three datasets with six common GNNs show that our framework can not only alleviate the bias between internal-links and cross-links but also boost the overall accuracy. Comparisons with other state-of-the-art methods also verify the superiority of our method.

## 1 Introduction

Recently, due to the strong capability in learning the latent representation of graph structure data, *Graph Neural Networks* (GNNs)-based link prediction methods have received increasing research interests and show excellent performance in recommendation systems [33, 34], bioinformatics [7, 16], and knowledge graph [24, 40]. However, these link prediction methods often prioritize performance without considering potential bias on the sensitive attributes of nodes, such as genders or regions, thus leading to social risks or the creation of informa-

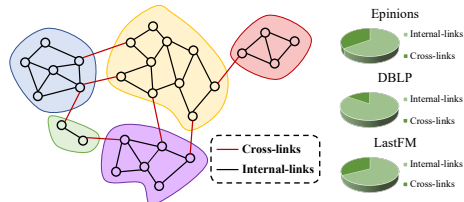


Figure 1: A toy graph for illustrating the concept of topological communities. The proportions of two kinds of links also are provided on the right.

\*Hong Huang is the corresponding author.

tion cocoons. For instance, in social recommendations, models tend to recommend users with friends belonging to the same region, limiting users’ opportunities to connect with the outside world [46]. Similarly, in political news recommendations, recommenders prefer to deliver content that aligns with users’ partisan beliefs, filtering out different perspectives and narrowing users’ scopes [23]. To tackle these issues, researchers have put forward several solutions. For example, FLIP [28] employs adversarial learning to encourage the model to predict outcomes that are independent of sensitive attributes. Similarly, CFC [4] aims to debias the node embeddings with compositional adversarial constraints. FairAdj [20] learns edge weights to generate a fair adjacency matrix, which is used for subsequent link prediction tasks. UGE [38] explores unbiased node embeddings from an unobserved graph, and proposes two kinds of methods based on reweighting and regularization constraints.

Although these methods make great progress in mitigating the bias in link prediction, there are two issues remaining unsolved: 1) All these methods generally modify the objective functions, like adding regularization constraints [4, 28] or reweighting [38] for mitigating the bias. However, the mechanism of simply modifying objective function may influence the optimization trajectory of the model and result in convergence towards a sub-optimal status. The experimental trade-off between debias and utility on multiple prior works [4, 20, 28, 38] also supports this claim. 2) Existing methods simply focus on sensitive attributes of nodes while ignoring the bias based on graph topology. In fact, due to GNNs’ heavy dependence on the aggregation of neighborhood messages, for the link prediction tasks, GNNs tend to wire new links inside the local community<sup>2</sup> (denoted as **internal-links**), while ignoring the links connecting different communities (denoted as **cross-links**), and this kind of bias may leave the graph in danger of being trapped in information cocoons or disconnected [21, 28], as shown in Figure 1. Empirical evidence can also be provided in Figure 2, in which we illustrate the top 10 commonest labels’ distributions in different communities. It can be seen that each community contains its own specific information pattern [26], making it challenging for one single community to encompass all diverse information in a network. This observation implies the propensity of graphs to form information cocoons with insufficient cross-links as bridges<sup>3</sup>. Motivated by these findings, we aim to design a GNN model that could boost the link prediction performance on cross-links and mitigate the bias between internal-links and cross-links without sacrificing utility.

By rethinking the problem of bias between internal-links and cross-links with a fresh insight from the data perspective, we statistics the proportion of internal-links and cross-links in three real-world datasets in Figure 1, and we can find that the number of internal-links far exceeds that of cross-links. This kind of data bias may be further enlarged and perpetuated by stacking GNN aggregation operations and finally leads to a biased link prediction. In light of this, we propose a simple yet effective twin-structure framework, which consists of two independent GNN models to mitigate the bias between internal-links and cross-links. Specifically, we first divide the whole graph into multiple communities that share similar topological locations and differentiate links into internal-links and cross-links. After that, in order to alleviate the data bias between internal-links and cross-links, supervision augmentation based on multiple rules is proposed to increase the supervised signals for cross-links, which could help the GNNs to better capture the patterns of cross-links and further generate debiased node embeddings. Subsequently, to avoid utility degradation, we design an embedding fusion module to merge original node embeddings and debiased node embeddings with a dynamic training strategy. In this way, the embedding fusion module could effectively preserve the performance of internal-links while alleviating the bias between internal-links and cross-links. Our main contributions can be summarized as follows:

- We reveal a significant link prediction bias based on the graph topology, i.e. the bias between internal-links and cross-links, and design a model-agnostic framework, which can help most of the GNNs address such bias without compromising utility.

<sup>2</sup>Generally, it refers to a cluster of nodes that enjoy similar topological location, which can be detected by community detection algorithms. In this paper, we deploy the Louvain algorithm [2] as the default detection method.

<sup>3</sup>Further analysis on elaborating the role of cross-links is provided in Section A of the Appendix.

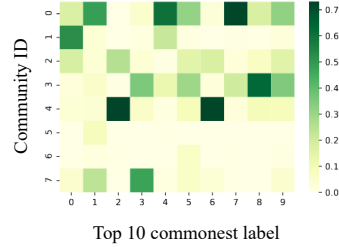


Figure 2: Distributions for top 10 labels in 10 different communities detected by Louvain on LastFM

- We propose three core components in our framework, including supervision augmentation, twins-GNN, and embedding fusion module. The combination of supervision augmentation and twins-GNN can generate debiased embeddings from a data aspect as demonstrations, while the embedding fusion module can effectively filter out the noise and preserve the utility of GNN.
- We evaluate our method on three real-world datasets with six common GNNs as base models and as well as compare them with other state-of-the-art baselines. The experimental results consistently demonstrate that our framework could effectively reduce the bias between internal-links and cross-links with even improved overall performance.

## 2 Related Work

**Link Prediction** As one of the most important applications in graph learning, link prediction algorithms have received extensive attention in the last two decades [3, 12, 14, 29]. The traditional link prediction methods are mainly based on some heuristic metrics on the graph structure, such as CN [25], AA [1], and Jaccard coefficient. In recent years, shallow embedding-based link prediction methods such as DeepWalk [29], Node2Vec [12], and LINE [36] have also emerged. With the powerful capability in representation learning on graph structure data, GNNs have achieved great progress in link prediction tasks as well [3, 13, 14, 19, 37, 48]. For example, LightGCN discovers the redundancy of non-linear activation and feature transformation and achieves better performance on recommendations [14]. PPRGo [3] uses a Personalized Page Rank matrix to efficiently approximate the multi-hop aggregation, which breaks the classical message-passing paradigm of GNNs. Following LightGCN, UltraGCN [27] approximates the infinite-layer aggregation for better recommendations. Besides, instead of learning from the whole graph, another line of research focuses on learning from relatively small subgraphs, such as SEAL [44] and SUREL+ [42], which also achieve remarkable performance in link prediction.

**Bias Related Issues in GNNs** Recently, bias-related issues have been widely concerned in graph neural networks, termed as "debias" or "fairness". Concentrating on node classification tasks, there is a branch of works on achieving fair results that are uncorrelated with sensitive attributes, including FairGNN [8], FairVGNN [39], EDITS [9], and EqGNN [32]. As for bias in link prediction, based on node2vec [12], Fairwalk [30] proposes a more fair random walk strategy, but fails to avoid the decrease in models' utility after achieving satisfactory fairness. Both FLIP [28] and CFC [4] try to mitigate the dyadic bias by proposing adversarial constraints in the loss function. Li et al. [20] designed a model named FairAdj, which could generate a fair adjacency matrix with different edge weights to address the dyadic bias with competitive utility. Similar to FairAdj, UGE [38] derives unbiased embeddings from an unobserved graph that involves no sensitive attribute information. However, all these methods modify the original objective functions and face a trade-off between debias and utility.

## 3 Methodologies

### 3.1 Problem Formulation

In this work, a graph is denoted by  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , and it consists of  $N = |\mathcal{V}|$  vertices and  $|\mathcal{E}|$  links. In the real world, nodes on the graph spontaneously form local communities [26], such as social circles in social networks, and various bundles of also-buy items on product graphs. We use  $\mathcal{C}(v) \in \mathbb{N}$  to denote the community membership of a given node  $v$  and define *cross-links* and *internal-links*:

**Definition 1. Cross-links and Internal-links.** Given a link  $\langle u, v \rangle$ , it belongs to cross-links  $\mathcal{E}_{cr}$  if its endpoints' memberships satisfy  $\mathcal{C}(u) \neq \mathcal{C}(v)$ ; otherwise, it belongs to internal-links  $\mathcal{E}_{in}$  as it satisfies  $\mathcal{C}(u) = \mathcal{C}(v)$ . So according to the endpoints' memberships, the links  $\mathcal{E}$  in a graph could be divided into  $\mathcal{E}_{in}$  and  $\mathcal{E}_{cr}$  two sets exclusively, i.e.  $\mathcal{E}_{in} \cap \mathcal{E}_{cr} = \emptyset$  and  $\mathcal{E}_{in} \cup \mathcal{E}_{cr} = \mathcal{E}$ .

Centering on the link prediction based on GNNs, each node will be mapped to an  $D$  dimensional embedding vector  $\mathbf{z} \in \mathbb{R}^D$  by a GNN encoder. The dot product score of a pair of nodes' embeddings implies the GNN model's confidence on whether there will be a potential link between two given vertices. Our goal is to mitigate the bias between internal-links and cross-links, i.e.:

**Definition 2. Bias between internal-links and cross-links (Bias).** Given a trained GNN model, a confidence matrix  $\mathbf{P} \in \mathbb{R}^{N \times N}$  can be inferred by calculating the dot product of endpoints'

embeddings, and a higher  $\mathbf{P}_{uv}$  indicates that model predicts that it very likely exists a link between node  $u$  and node  $v$ . Let the notation  $\mathcal{M}$  represent an evaluation metric in link prediction tasks, and we use the difference between two kinds of links' average performance to define the bias:

$$\text{Bias} = |\mathbb{E}[\mathcal{M}(\mathbf{P})|\mathcal{E} = \mathcal{E}_{in}] - \mathbb{E}[\mathcal{M}(\mathbf{P})|\mathcal{E} = \mathcal{E}_{cr}]| \quad (1)$$

**Task definition.** To address such bias, a GNN recommender should balance the performance on internal-links and cross-links, so that the final model could show equivalent capability on mining true internal-links and cross-links. The goal of this paper is to design a model-agnostic framework, which can enhance the embedding ability of a given GNN model so that the bias metrics are reduced while the accuracy metrics are at least not decreased:

$$\text{Bias}(\mathcal{M}(\mathbf{P}|\mathbf{Z})) < \text{Bias}(\mathcal{M}(\mathbf{P}|\mathbf{Z}')), \quad \mathcal{M}(\mathbf{P}|\mathbf{Z}) \geq \mathcal{M}(\mathbf{P}|\mathbf{Z}') \quad (2)$$

where  $\mathbf{Z}'$  represents the original node embeddings given by a GNN, and  $\mathbf{Z}$  represents the node embeddings enhanced by our framework.

### 3.2 Framework Overview

We propose a twin-structure GNN-enhancement framework, which can reduce the bias between cross-links and internal-links without hurting the overall performance. An overview of our framework is illustrated in Figure 3. Its key components include supervision augmentation, Twins GNNs, and an embedding fusion module. Details will be provided in the following sections. In traditional graph embedding models, existing links on the graph are used as supervision signals. In our framework, we first generate a certain number of pseudo cross-links and form the augmented supervision signals. Then, we use two GNNs with the same architecture to model the two sets of supervision signals (original supervision and augmented supervision) independently, leading to original node embeddings and debiased node embeddings. Lastly, to avoid the impact of noise during supervision augmentation, an embedding fusion module is proposed, which ensures that the final embeddings will retain the merits of both original node embeddings and debiased node embeddings.

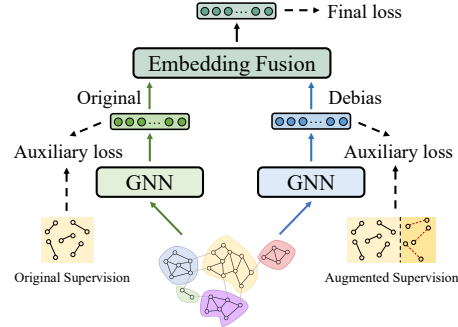


Figure 3: The overview of our framework

### 3.3 Supervision Augmentation

As revealed in Figure 1, in most datasets, usually the vast majority of links on the graph are internal-links, and this kind of disparity may cause the learned model to be biased to internal-links while neglecting the performance of the cross-links in order to obtain better overall metrics. Therefore, we design an augmentation step to alleviate the supervision signal sparsity issue.

**Jaccard based augmentation.** Inspired from the concept of edge strength [11], a set of  $K$  pairs of nodes  $\{\langle \hat{u}_1, \hat{v}_1 \rangle, \dots, \langle \hat{u}_K, \hat{v}_K \rangle\}$ , which satisfy  $\mathcal{C}(\hat{u}_i) \neq \mathcal{C}(\hat{v}_i), \forall i \in \{1, \dots, K\}$ , and have the top  $K$  highest Jaccard coefficient score  $\mathcal{J}(\hat{u}, \hat{v})$ , are wired as pseudo cross-links. Specifically, the  $\mathcal{J}(\hat{u}, \hat{v})$  can be formulated as:

$$\mathcal{J}(\hat{u}, \hat{v}) = \frac{|\mathcal{N}(\hat{u}) \cap \mathcal{N}(\hat{v})|}{|\mathcal{N}(\hat{u}) \cup \mathcal{N}(\hat{v})|} \quad (3)$$

and  $K = k * (|\mathcal{E}_{in}| - |\mathcal{E}_{cr}|)$ , where  $k$  is a hyper-parameter to control the supervision augmentation size  $K$ , and  $\mathcal{N}(\cdot)$  represents the neighbor nodes of a given node.  $\mathcal{J}(\cdot, \cdot)$  measures the proportion of common neighbors between two nodes, and intuitively, a pair of nodes with more common neighbors are more likely to form a potential link, which ensures the high confidence of augmented signals.

**Random walk based augmentation.** However, when a network is extremely dense, adopting Jaccard-based augmentation will consume too much computing resource due to the "neighbor explosion" issue. What's more, the Jaccard-based supervision augmentation may easily only cover nodes around the border of communities, and nodes in the center of a community may still fail to benefit from the

augmentation. To solve these problems, we propose a new supervision augmentation method based on random walks. Specifically, we take random walks in the original graph, and each pair of nodes  $\langle \hat{u}, \hat{v} \rangle$  that appear in the same walk path will be recorded. Next, cross-community node pairs, *i.e.*,  $\mathcal{C}(\hat{u}) \neq \mathcal{C}(\hat{v})$ , with the top  $K$  highest co-occurrence frequency will be screened out as augmented supervision signals. Similarly,  $K = k * (|\mathcal{E}_{in}| - |\mathcal{E}_{cr}|)$  represents the supervision augmentation size.

Note that, we only generate pseudo supervision signals, and the original graph structure, which is used for message passing in GNNs, remains unchanged. We believe that the lack of supervision plays a much more important role in causing the bias between internal-links and cross-links and adding supervision signals is a more straightforward way to alleviate this kind of bias. Let  $\mathcal{E}^P$  denote the newly constructed edge set. The original supervision signals, which are the original links on the graph, are denoted by  $\mathcal{E}^O$ , and the final augmented supervision signals are denoted by  $\mathcal{E}^A = \mathcal{E}^O \cup \mathcal{E}^P$ .

### 3.4 Twins GNNs

Recently, the retrieval-based learning paradigm, in which the core idea is to provide supporting examples that can be used as references for the model, has extracted increasing research interest in the field of natural language processing [35]. Inspired by this, it would be a promising way to generate debiased node embeddings as references for our framework to mitigate the bias between internal-links and cross-links. Specifically, we let  $\mathcal{E}^O$  and  $\mathcal{E}^A$  to guide the training of two twin GNNs, respectively. The two twin GNNs share the same model architecture but have independent sets of parameters. To this end, the twin-structure GNNs could generate two kinds of embeddings named original node embeddings  $\mathbf{Z}^O$  and debiased node embeddings  $\mathbf{Z}^A$ , respectively. In order to make sure that the twin-structure GNNs could precisely generate original embeddings and debiased embeddings for further embedding fusion, we follow the literature [14, 34] and employ one of the most classic loss functions in recommendation systems - BPRLoss to design our two auxiliary loss functions for the twin-structure GNNs, which can be generalized as:

$$\mathcal{L} = -\frac{1}{|\mathcal{E}|} \sum_{\langle u, v \rangle \in \mathcal{E}} \log \sigma(r_{u,v} - r_{u, \hat{v}}) \quad (4)$$

where  $\mathcal{E}$  could be  $\mathcal{E}^O$  or  $\mathcal{E}^A$  depending on the type of generated embeddings, and the auxiliary loss functions, are denoted as  $\mathcal{L}^O$  and  $\mathcal{L}^A$ , respectively.  $\sigma$  is an activation function like sigmoid.  $\langle u, \hat{v} \rangle$  denotes a negative sample, and  $r_{u,v}$  denotes the prediction score of a given node pair  $\langle u, v \rangle$ . Note that, other alternative loss functions with multiple negative samples can be also easily deployed to our method, which is determined by the practitioners according to their practical applications. Intuitively, with the balanced proportion of two types of links in  $\mathcal{E}^A$  after supervision augmentation, the debiased node embeddings  $\mathbf{Z}^A$  could show excellent capability in eliminating the bias between internal-links and cross-links. The debiased node embeddings  $\mathbf{Z}^A$  will become a good demonstration for subsequent embedding fusion modules in the further learning of final embeddings.

### 3.5 Embedding Fusion

Although we set several heuristic rules in Section 3.3 to ensure that the generated signals have high confidence, the supervision augmentation stage will inevitably introduce noise. Thus,  $\mathbf{Z}^A$  is trained with impure supervision. If fusing  $\mathbf{Z}^A$  into  $\mathbf{Z}^O$  in a naive way, such as a simple averaging, although the performance on the cross-links can be improved, the internal-links' will be considerably degraded.

Thus, we adopt an embedding fusion module  $F(\cdot)$  that merges two kinds of embeddings, *i.e.* original node embeddings  $\mathbf{Z}^O$  and debiased node embeddings  $\mathbf{Z}^A$ , to refine meaningful information from the debiased embeddings and fuse them into the original embeddings:

$$\mathbf{H}_i = \text{CONCAT}(\mathbf{Z}_i^O, \mathbf{Z}_i^A) \quad (5)$$

$$\mathbf{Z}_i = F(\mathbf{H}_i, \theta^F) = \mathbf{W}_2 \sigma(\mathbf{W}_1 \mathbf{H}_i + \mathbf{b}_1) + \mathbf{b}_2 \quad (6)$$

where  $\mathbf{W}_1 \in \mathbb{R}^{D \times 2D}$ ,  $\mathbf{b}_1 \in \mathbb{R}^D$ ,  $\mathbf{W}_2 \in \mathbb{R}^{D \times D}$ ,  $\mathbf{b}_2 \in \mathbb{R}^D$  are the trainable parameters of  $F(\cdot)$ , and can be abbreviated as  $\theta^F$  for conciseness.  $\sigma$  is the activation function and  $D$  is the embedding dimension. Note that, both  $\mathbf{Z}_i^O$  and  $\mathbf{Z}_i^A$  are output embeddings from GNN models supervised by auxiliary loss, and we believe that there is no need to further deploy a complex network during embedding fusion, hence we simply show a 2-layer MLP in Eq.(6).

As the core component of our framework, we argue that the final embeddings  $\mathbf{Z}$  after embedding fusion function  $F(\cdot)$  should filter signal noise in  $\mathcal{E}^A$  while preserving as much correct augmented information as possible. Therefore, we optimize embedding fusion module  $F(\cdot)$  and twin-GNNs by minimizing the following objective function :

$$\mathcal{L}^F = -\frac{1}{|\mathcal{E}^O|} \sum_{\langle u,v \rangle \in \mathcal{E}^O} \log \sigma(r_{u,v} - r_{u,\hat{v}}) \quad (7)$$

---

**Algorithm 1** Proposed training process

---

**Input:** Graph  $\mathcal{G}$  with link set  $\mathcal{E}^O$ . Hyper-parameters:  $\alpha$ ,  $\beta$ , and  $T$ , learning rate  $\gamma^O, \gamma^A$ .  
**Output:** Twin GNNs' parameters  $\theta^O, \theta^A$ , and embedding fusion module parameters  $\theta^F$ .  
1: Randomly initialize twins GNN models with  $\theta^O, \theta^A$ , embedding fusion module with  $\theta^F$ .  
2: Split  $\mathcal{G}$  into  $|\mathcal{C}|$  communities and categorize links into internal-links and cross-links.  
3: Select augmented supervision signals  $\mathcal{E}^A$  by the Jaccard coefficient or co-occurrence frequency.  
4: **while** not converged **do**  
5:   Compute  $\mathcal{L}^O$  and  $\mathcal{L}^A$  by Eq.(4)  
6:   Update twins GNN models:  $\theta^O \leftarrow \theta^O + \gamma^O \cdot \nabla_{\theta^O} \mathcal{L}^O$ ,  $\theta^A \leftarrow \theta^A + \gamma^A \cdot \nabla_{\theta^A} \mathcal{L}^A$   
7:   Compute learning rate  $\gamma_t^F$  and step size  $S_t$  by Eq.(8)  
8:   **for**  $step = 1$  **to**  $S_t$  **do**  
9:     Compute  $\mathcal{L}^F$  by Eq.(7)  
10:     Update embedding fusion module:  $\theta^F \leftarrow \theta^F + \gamma_t^F \cdot \nabla_{\theta^F} \mathcal{L}^F$   
11:     Update twins GNN models:  $\theta^O \leftarrow \theta^O + \gamma_t^F \cdot \nabla_{\theta^O} \mathcal{L}^F$ ,  $\theta^A \leftarrow \theta^A + \gamma_t^F \cdot \nabla_{\theta^A} \mathcal{L}^F$   
12:   **end for**  
13: **end while**  
14: **return**  $\theta^O, \theta^A, \theta^F$

---



---

**Algorithm 2** Proposed inference process

---

**Input:** Graph  $\mathcal{G}$  with link set  $\mathcal{E}^O$ . Trained Twin GNNs' parameters  $\theta^O, \theta^A$ , and embedding fusion module parameters  $\theta^F$   
1: Get node embeddings  $Z^O$  through parameters  $\theta^O$   
2: Get node embeddings  $Z^A$  through parameters  $\theta^A$   
3: Get node embeddings  $Z$  through combining  $Z^O$  and  $Z^A$  with embedding fusion module  $\theta^F$   
4: Evaluation through test node pairs with node embeddings  $Z$

---

### 3.6 Implementation Algorithms

For a better understanding of our method, here we summarize the detailed training process and inference process, and the pseudo codes are provided in Algorithm 1 and Algorithm 2, respectively.

**Training process:** Firstly, during the preprocessing stage,  $C = |\mathcal{C}|$  communities are detected through Louvain algorithm [2] and all links are categorized into internal-links or cross-links, respectively (line 2). In addition, newly constructed supervision signals  $\mathcal{E}^P$  are also generated during supervision augmentation (line 3). After that, original node embeddings  $\mathbf{Z}^O$  and debiased node embeddings  $\mathbf{Z}^A$  are generated by minimizing the auxiliary loss functions in Section 3.4 (lines 5-6) with learning rate  $\gamma^O$  and  $\gamma^A$ , respectively. The two sets of generated embeddings will be forwarded to the embedding fusion module  $F(\cdot)$  to get the final embeddings  $\mathbf{Z} = F(\mathbf{Z}^O, \mathbf{Z}^A)$ , and minimize the final loss function with multiple training steps (lines 8-12.) Essentially, the embedding fusion module  $F(\cdot)$  is highly related to the quality of original embeddings  $\mathbf{Z}^O$  and debiased embeddings  $\mathbf{Z}^A$ , thus we introduce a dynamic training strategy to avoid excessive training for embedding fusion module when  $\mathbf{Z}^O$  and  $\mathbf{Z}^A$  have not been stable yet. Specifically, the learning rate  $\gamma_t^F$  and training step  $S_t$  for  $F(\cdot)$  at epoch  $t$  is:

$$\gamma_t^F = \alpha * \frac{1}{1 + e^{-(t-T)}} \quad , \quad S_t = \beta * \frac{1}{1 + e^{-(t-T)}} \quad (8)$$

where  $\alpha$ ,  $\beta$ , and  $T$  are hyper-parameters to control the learning rate, training steps, and the epoch that  $F(\cdot)$  can believe the quality of input embeddings have been stable, respectively.

**Inference process:** During the inference stage, we use the node embeddings  $\mathbf{Z}$  for evaluation (line 4), which are the output of the embedding fusion module. Specifically, we first get node embeddings  $\mathbf{Z}^O$  and  $\mathbf{Z}^A$  through the trained twins GNN in advance (lines 1-2), and then perform embedding fusion operation (line 3) to get the final node embeddings  $\mathbf{Z}$ .

## 4 Experiment

### 4.1 Experimental Settings

**Datasets.** We conduct our experiments on three real-world networks from SNAP<sup>4</sup> and RecBole<sup>5</sup> [47] public datasets. Specifically, to verify our framework’s extendibility, we choose two kinds of networks according to the types of interaction between nodes, including user-user (Epinions, DBLP), and user-item (LastFM). **Epinions** is a who-trust-whom social network extracted from the online review site Epinions.com. Each node represents a consumer, and each directed link represents a consumer-to-consumer trust connection. **DBLP** is a co-author social network, which is collected from the DBLP computer science bibliography, and two author nodes are connected if they used to publish at least one paper together. **LastFM** dataset contains users’ listening information from the Last.fm online music system and each listening event represents a user-artist interaction. The detailed statistics for these three datasets are reported in Table 1, and other experimental settings on datasets are described in Section B in the Appendix.

Table 1: Dataset Statistics

Datasets	Users	Items	Interactions	Density	Type
Epinions	75,879	-	508,837	0.000088	User-user
DBLP	317,080	-	1,049,866	0.000010	User-user
LastFM	1,892	17,632	92,834	0.002783	User-item

**Base models.** Because our proposed method is a model-agnostic framework, which is compatible with almost all GNN-based link prediction models, here we choose six common and effective GNNs as our base models, including **GraphSAGE** [13], **GIN** [41], **GAT** [37], **LightGCN** [14], **PPRGo** [3], and **UltraGCN** [27]. We verify if our framework could mitigate the bias between internal-links and cross-links without hurting the original overall performance.

**Reproducibility.** All experiments are run on machines with the same configuration: Intel(R) Xeon(R) CPU E5-2680 and Tesla V100 GPU with 32GB memory. How to detect the communities of a network is not the research focus of this paper, and here we apply the Louvain algorithm [2] to detect the communities in a graph after considering its high speed and effectiveness. Other alternative community detection algorithms and hyper-parameters will be discussed in the Appendix. The source code and data are available at <https://github.com/CGCL-codes/Cross-links-Bias>.

### 4.2 Main Results

In this part, we show the performance of our method in terms of both link prediction utility and debias effectiveness after applying it to various GNN models.

The main results on three datasets are reported in Table 2. For Epinions and DBLP, we use Jaccard-based augmentation here, and considering the high density of LastFM, we use random walk-based augmentation. More results with random walk-based augmentation on Epinions and DBLP are listed in Section C in the Appendix for reference. The internal-links performance (Internal.), cross-links performance (Cross.), and overall performance (Overall) are calculated on internal-links in the test set, the cross-links in the test set, and the whole test set respectively. To verify our method’s capability of addressing the bias, *Bias* in Eq.(1) is also computed to evaluate the performance difference between internal-links and cross-links. Table 2 shows the results of our approach and base models, and we can have the following observations:

- The performance of internal-links outperforms that of cross-links under all settings, which indicates the link prediction bias between internal-links and cross-links is widespread. Note that, after proposing our framework, we can only ease this kind of bias to some extent, which inspires us the reasons for the poor performance on cross-links may not only come from the data perspective, and we leave this problem as a future work.

<sup>4</sup><https://snap.stanford.edu/data/index.html>

<sup>5</sup><https://recbole.io/index.html>

Table 2: Link prediction performance (Hits@50) of internal-links, cross-links, and the whole link set of our methods and corresponding base models on three real-world datasets. The results are reported in percentage (%). We **bold** the results when our framework improves the base GNN model.

		Epinions				DBLP				LastFM			
		Internal.↑	Cross.↑	Overall↑	Bias↓	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	32.84	14.84	26.80	18.00
	Debias	<b>36.98</b>	<b>34.45</b>	<b>36.08</b>	<b>1.63</b>	<b>77.18</b>	<b>31.54</b>	<b>66.23</b>	<b>45.64</b>	32.73	<b>15.13</b>	<b>26.80</b>	<b>17.60</b>
GIN	Orig.	33.49	30.97	32.59	2.52	56.66	16.86	47.29	39.80	33.63	16.91	28.00	16.72
	Debias	<b>40.56</b>	<b>39.39</b>	<b>40.20</b>	<b>1.26</b>	<b>60.61</b>	<b>34.03</b>	<b>54.35</b>	<b>26.58</b>	32.13	<b>19.88</b>	<b>28.00</b>	<b>12.25</b>
GAT	Orig.	39.30	34.90	37.73	4.40	66.25	22.47	55.94	43.78	32.28	12.76	25.70	19.52
	Debias	<b>39.58</b>	<b>36.13</b>	<b>38.35</b>	<b>3.45</b>	<b>75.66</b>	<b>37.46</b>	<b>66.66</b>	<b>38.20</b>	<b>34.24</b>	<b>16.32</b>	<b>28.20</b>	<b>17.92</b>
PPRGo	Orig.	42.86	28.75	37.83	14.11	85.71	41.14	75.28	44.58	34.84	17.51	29.00	17.33
	Debias	<b>47.44</b>	<b>42.34</b>	<b>45.62</b>	<b>5.10</b>	<b>90.54</b>	<b>55.64</b>	<b>82.32</b>	<b>34.90</b>	<b>35.14</b>	<b>19.88</b>	<b>30.00</b>	<b>15.26</b>
LightGCN	Orig.	46.43	37.11	43.11	9.32	85.95	47.41	76.88	38.54	38.16	16.62	30.90	21.54
	Debias	<b>51.49</b>	<b>45.31</b>	<b>49.29</b>	<b>6.18</b>	<b>93.55</b>	<b>65.33</b>	<b>86.90</b>	<b>28.22</b>	<b>38.31</b>	<b>17.51</b>	<b>31.30</b>	<b>20.80</b>
UltraGCN	Orig.	30.62	5.81	21.78	24.81	95.74	63.82	88.22	31.92	32.73	15.73	27.10	17.00
	Debias	<b>44.13</b>	<b>43.66</b>	<b>43.96</b>	<b>0.67</b>	<b>97.14</b>	<b>71.71</b>	<b>91.15</b>	<b>25.43</b>	<b>35.29</b>	<b>18.40</b>	<b>29.60</b>	<b>16.89</b>

- After proposing our method, all base models get improved on both internal-links and cross-links in most cases, which results in an improvement in the overall performance. The reasons for this impressive improvement are two-fold. For one thing, supervision augmentation directly helps boost the performance of cross-links. For another, the augmented supervisions play a regularization role in the learning of node embeddings, thus, the overall quality of representations can be improved.
- The increase of cross-links is much higher than that of internal-links, and this contributes to the decrease of bias. For instance, when our framework is applied to LightGCN on DBLP, the performance of internal-links improves from 85.95% to 93.55%, and the performance of cross-links improves from 47.41% to 65.33%, thus the bias decreases from 38.54% to 28.22%.

### 4.3 Ablation Study

In this section, we want to explore: 1) the impact of supervision augmentation on our framework; 2) the impact of the embedding fusion module on our framework; 3) the impact of two auxiliary loss functions in Eq.(4); 4) the impact of dynamic training strategy during training embedding fusion module. As an example, we show the results of LightGCN on two datasets in Table 3. All results are based on Jaccard-based augmentation.

We first replace the augmented cross-links signals, which are selected through

Jaccard coefficient, with random node pairs across different communities, and denote this variant as "-Augment". There are two main observations: 1) After adding random-augmented signals, the overall performance still shows a significant improvement compared to the base model, which indicates that the embedding fusion module can effectively filter out useful information from the augmented embeddings and fuse it into the original embeddings; 2) Compared with our framework, the random-augmented framework demonstrates unsatisfactory results on reducing the performance disparity between cross-links and internal-links, such as slightly reducing this kind of bias on Epinions dataset from 9.32% (46.43% - 37.11%) to 8.46% (49.43% - 40.97%), and this indicates the importance of our supervision augmentation method on debias.

Secondly, after removing the embedding fusion module (denoted as "-Fusion"), although the cross-links performance of LightGCN gets significant improvement, i.e., from 37.11% to 46.39% on the Epinions dataset and from 47.41% to 66.97% on the DBLP dataset, the internal-links performance and overall performance get severe influence comparing with our framework. Especially on DBLP, the internal-links performance of LightGCN is deceased by 13.03% (from 85.95% to 72.92%), while our framework could even slightly improve the performance of internal-links. It indicates that, after proposing the embedding fusion module, our framework could effectively filter out noisy augmented signals to avoid performance deterioration.

Table 3: Ablation studies with LightGCN as the base model. The results (Hits@50) are reported in percentage (%). The best results are **bold**, and the runner-up is underlined.

	Epinions			DBLP		
	Internal.↑	Cross.↑	Overall↑	Internal.↑	Cross.↑	Overall↑
Original	46.43	37.11	43.11	85.95	47.41	76.88
Ours	<b>51.49</b>	<u>45.31</u>	<b>49.29</b>	<b>93.55</b>	<u>65.33</u>	<b>86.90</b>
- Augment	49.43	40.97	<u>46.77</u>	92.76	60.97	<u>85.24</u>
- Fusion	44.31	<b>46.39</b>	44.87	72.92	<b>66.97</b>	71.53
- Auxiliary Loss	<u>49.63</u>	41.55	46.75	85.31	60.61	84.85
- Dynamic	48.63	41.58	46.12	<u>93.09</u>	59.08	85.08



Finally, after discarding the two auxiliary loss functions of twin GNNs in Eq. (4) ("-Auxiliary Loss") or setting a fixed learning rate  $\gamma^{F*}$  and training step  $S^*$  in the embedding fusion module ("-Dynamic"), all metrics deteriorate slightly compared to our proposed framework. This matches our expectation, because both the lack of guidance from auxiliary loss and a fixed training strategy would inevitably influence the quality of two sets of input embeddings, thus influencing the final embeddings.

#### 4.4 Comparison with Other Competitors

In this section, we aim to compare our methods with several powerful competitors on reducing the bias between internal-links and cross-links. Specifically, we choose five highly related methods as our baselines, including FairWalk [30], CFC [4], FairAdj [20], FLIP [28], and UGE [38], and we replace the required sensitive attributes in these methods with community memberships detected by Louvain algorithm [2] in advance. To

be fair, all models' embedding dimensions are set to 64, and other details on hyper-parameters are listed in Section B in the Appendix for saving space. Note that, due to the algorithm's requirement on the multiplication of the adjacency matrix, FairAdj reports "OOM" errors when deployed on Epinions and DBLP. For comparison, we report the results of GraphSAGE [13] trained with our framework (denoted as Debias-SAGE). The same as Section 4.2, we take Jaccard-based augmentation on Epinions and DBLP and take random walk-based augmentation on LastFM.

As is shown in Table 4, our method achieves state-of-the-art results on overall performance with top-ranking debias results. Especially on the DBLP dataset, our method's overall performance far exceeds that of all other baselines, while achieving the best results in reducing the bias between internal-links and cross-links. We believe the reasons are two-fold: 1) With the model-agnostic design of our framework, we could easily deploy our method on many powerful GNN models. 2) Instead of adding extra constraints in the loss function or modifying the objective functions, our method aims at addressing the bias between internal-links and cross-links from a data perspective, and the supervision augmentation plays a regularization role in boosting the overall performance.

#### 4.5 Empirical Findings on Easing Information Cocoons

With emphasizing the performance on cross-links, one merit of our model is the potential capability to ease information cocoons, which can be empirically verified from the following two aspects.

**Debiased Recommendation.** In this part, we further explicitly reveal whether our framework can provide debiased recommendations for nodes to ease the information cocoons. In detail, we train several powerful GNN models, including LightGCN [14], PPRGo [3], and UltraGCN [27] on Epinions and DBLP. Next, we randomly select 2000 source nodes and observe the average proportion of internal-links in their historical data and their recommendation lists given by GNNs. As is shown in Figure 4, the recommendation lists given by GNN models significantly magnify the nodes' original preference on internal-links. From the perspective of the users (or nodes receiving recommendations), such biased recommendations will distort users' true preference by ignoring their niche interests (cross-links) while emphasizing the mainstream interests (internal-links), and make users feel confined to limited domains. For comparison, we provide the results of our enhanced GNN models as well, and it can be seen that after proposing our method (denoted as "Ours"), the distributions of

Table 4: Comparison with several competitors on both utility and debias. The average results (in Hits@50) are reported in percentage (%) after repeating each method three times. The best results are **bold**, and the runner-up is underlined.

	Epinions		DBLP		LastFM	
	Overall $\uparrow$	Bias $\downarrow$	Overall $\uparrow$	Bias $\downarrow$	Overall $\uparrow$	Bias $\downarrow$
FairWalk	23.75 $\pm$ 1.1	7.16 $\pm$ 0.7	48.81 $\pm$ 1.3	50.00 $\pm$ 2.1	23.05 $\pm$ 0.6	21.78 $\pm$ 1.0
CFC	27.35 $\pm$ 0.4	2.01 $\pm$ 0.3	54.12 $\pm$ 2.1	49.37 $\pm$ 1.6	24.28 $\pm$ 0.5	17.79 $\pm$ 0.3
FairAdj	—	—	—	—	23.97 $\pm$ 1.2	19.05 $\pm$ 0.9
FLIP	25.65 $\pm$ 0.6	4.65 $\pm$ 0.5	55.23 $\pm$ 1.3	<u>45.93</u> $\pm$ 1.1	22.73 $\pm$ 0.4	18.23 $\pm$ 0.5
UGE	27.22 $\pm$ 0.9	<b>1.20</b> $\pm$ 0.2	54.66 $\pm$ 1.8	54.36 $\pm$ 2.3	25.50 $\pm$ 0.9	<b>14.00</b> $\pm$ 0.7
GraphSAGE	30.69 $\pm$ 0.9	2.77 $\pm$ 0.3	<u>56.41</u> $\pm$ 1.5	54.65 $\pm$ 1.6	<u>26.80</u> $\pm$ 0.5	18.00 $\pm$ 0.6
Debias-SAGE	<b>36.08</b> $\pm$ 0.7	<u>1.63</u> $\pm$ 0.2	<b>66.23</b> $\pm$ 1.9	<b>45.64</b> $\pm$ 1.7	<b>26.80</b> $\pm$ 0.4	<u>17.60</u> $\pm$ 0.3

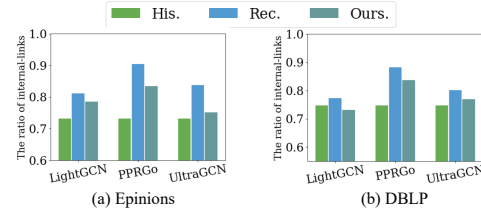


Figure 4: The internal-links proportion of historical distribution and multiple GNNs' recommendations before and after applying our method. "His.", "Rec.", and "Ours." denote the historical interactions, original recommendations, and our debiased recommendations, respectively.

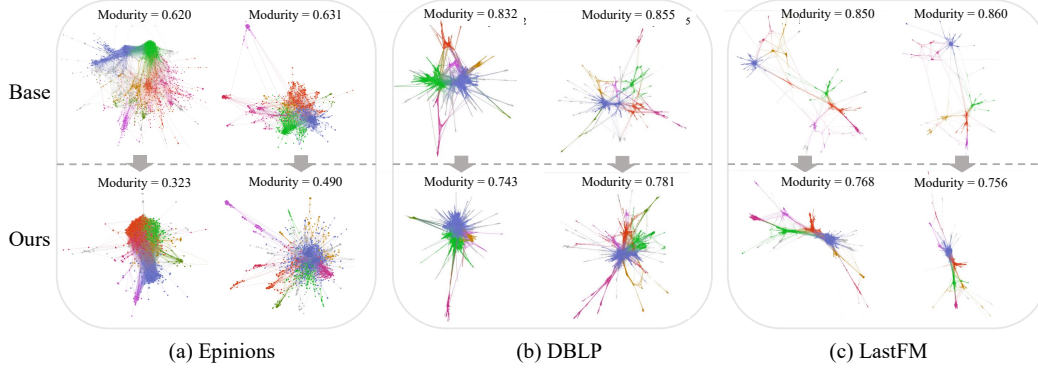


Figure 5: The visualization of subgraphs reconstructed by the embeddings learned through the base model (above) and ours (below). Note that, each column shows the visualization of the same subgraph, and subgraphs generated by our approach show minor topological isolation phenomena.

recommendations become closer to that of historical data, which indicates that our approach could effectively address GNN models’ biased recommendation and thus ease the information cocoons.

**Visualization of Breaking Filter Bubbles.** To visualize our proposed framework’s effect on alleviating information cocoons, we use Gephi<sup>6</sup> to show the graphs reconstructed by node embeddings learned through the base model and our model, respectively. To be specific, we take UltraGCN [27] as the base model here and only a subset of nodes from different communities are shown for conciseness. Note that, all the hyper-parameters in the Gephi platform are set to be the same, and the visualization results reflect the natural layout of the reconstructed networks. The visualization results are shown in Figure 5, and node colors represent the community tags. To better understand the visualization results, we also provide the modularity value in each subgraph, which measures the strength of the division of a network. It can be seen that after proposing our framework, except for some extreme outliers, the connections between most communities are becoming closer, and the isolation between communities is broken down. The reduced modularity values can also support our claims. These results indicate that the embedding generated by our model can effectively facilitate the generation of cross-links in a network, thus alleviating information cocoons.

## 5 Conclusion

In this paper, we aim to address the bias in GNN-based link prediction, especially the bias based on cross-links (links cross communities) after considering its specific roles in easing information cocoons and connecting different communities. We further break the paradigm of modifying objective functions in prior works while pursuing debiased performance, and turn to address the bias issue on cross-links from a data perspective, which could effectively alleviate the bias by even boosting models’ prediction utility. Specifically, by borrowing the concept in retrieval-based learning paradigms, debiased node embeddings are generated as demonstrations. An embedding fusion module with dynamic training strategies is also proposed to ensure that the final embeddings could retain the merits of both original embeddings and debiased embeddings. Experiments on three real-world datasets with six base GNNs indicate that our framework could not only reduce the performance disparity between internal-links and cross-links but also significantly improve the overall performance, and further alleviate the potential information cocoons.

**Limitations & Discussions.** Firstly, although we have greatly reduced the bias between internal-links and cross-links, the experimental results indicate that the bias is not clearly eliminated, which implies that data bias may not be the only reason. What’s more, despite the supervision augmentation having a potential regularization effect, and leading to improved performance on internal-links, we currently lack theoretical underpinnings and rigorous analysis for this phenomenon. However, we find some potential connections between our work and counterfactual learning. In our settings, supervision augmentation will introduce plenty of unobserved/counterfactual samples for GNN learning. This kind of counterfactual learning may help GNN to better capture the intrinsic relationships between a pair of nodes, thus giving a more accurate prediction.

<sup>6</sup><https://gephi.org/>

## Acknowledgement

The work is supported by the National Natural Science Foundation of China (No.62172174, No.61932004).

## References

- [1] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.
- [2] 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.
- [3] 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 *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 2464–2473. ACM, 2020.
- [4] A. J. Bose and W. L. Hamilton. Compositional fairness constraints for graph embeddings. In *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.
- [5] T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.
- [6] U. Chitra and C. Musco. Analyzing the impact of filter bubbles on social network polarization. In *Proceedings of the 13th ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pages 115–123. ACM, 2020.
- [7] M. Coşkun and M. Koyutürk. Node similarity-based graph convolution for link prediction in biological networks. *Bioinformatics*, 37(23):4501–4508, 2021.
- [8] E. Dai and S. Wang. Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021*, pages 680–688. ACM, 2021.
- [9] Y. Dong, N. Liu, B. Jalaian, and J. Li. EDITS: modeling and mitigating data bias for graph neural networks. In *Proceedings of the ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 1259–1269. ACM, 2022.
- [10] 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.
- [11] M. S. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78(6):1360–1380, 1973.
- [12] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 855–864. ACM, 2016.
- [13] W. L. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, NeurIPS 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1024–1034, 2017.
- [14] 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.

- [15] 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 *Proceedings of the Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.
- [16] K. Jha, S. Saha, and H. Singh. Prediction of protein–protein interaction using graph neural networks. *Scientific Reports*, 12(1):8360, 2022.
- [17] 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.
- [18] 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.
- [19] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017*, OpenReview.net, 2017.
- [20] P. Li, Y. Wang, H. Zhao, P. Hong, and H. Liu. On dyadic fairness: Exploring and mitigating bias in graph connections. In *Proceedings of the 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.
- [21] Z. Li, Y. Dong, C. Gao, Y. Zhao, D. Li, J. Hao, K. Zhang, Y. Li, and Z. Wang. Breaking filter bubble: A reinforcement learning framework of controllable recommender system. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pages 4041–4049. ACM, 2023.
- [22] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Comput.*, 7(1):76–80, 2003.
- [23] P. Liu, K. Shivaram, A. Culotta, M. A. Shapiro, and M. Bilgic. The interaction between political typology and filter bubbles in news recommendation algorithms. In *Proceedings of the Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 3791–3801. ACM / IW3C2, 2021.
- [24] S. Liu, B. C. Grau, I. Horrocks, and E. V. Kostylev. INDIGO: gnn-based inductive knowledge graph completion using pair-wise encoding. In *Proceedings of the Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 2034–2045, 2021.
- [25] F. Lorrain and H. C. White. Structural equivalence of individuals in social networks. *The Journal of Mathematical Sociology*, 1(1):49–80, 1971.
- [26] Z. Luo, J. Lian, H. Huang, H. Jin, and X. Xie. Ada-gnn: Adapting to local patterns for improving graph neural networks. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*, pages 638–647. ACM, 2022.
- [27] K. Mao, J. Zhu, X. Xiao, B. Lu, Z. Wang, and X. He. Ultragcn: Ultra simplification of graph convolutional networks for recommendation. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 1253–1262. ACM, 2021.
- [28] F. Masrour, T. Wilson, H. Yan, P. Tan, and A. Esfahanian. Bursting the filter bubble: Fairness-aware network link prediction. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 841–848. AAAI Press, 2020.
- [29] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 701–710. ACM, 2014.

- [30] T. Rahman, B. Surma, M. Backes, and Y. Zhang. Fairwalk: Towards fair graph embedding. In *Proceedings of the 38th International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3289–3295. International Joint Conferences on Artificial Intelligence Organization, 7 2019.
- [31] Y. Shen, Y. Wu, Y. Zhang, C. Shan, J. Zhang, K. B. Letaief, and D. Li. How powerful is graph convolution for recommendation? In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 1619–1629. ACM, 2021.
- [32] U. Singer and K. Radinsky. Eqgnn: Equalized node opportunity in graphs. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence, AAAI 2022, February 22 - March 1, 2022*, pages 8333–8341. AAAI Press, 2022.
- [33] X. Song, J. Lian, H. Huang, Z. Luo, W. Zhou, X. Lin, M. Wu, C. Li, X. Xie, and H. Jin. xgc: An extreme graph convolutional network for large-scale social link prediction. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pages 349–359. ACM, 2023.
- [34] X. Song, J. Lian, H. Huang, M. Wu, H. Jin, and X. Xie. Friend recommendations with self-rescaling graph neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pages 3909–3919. ACM, 2022.
- [35] Y. Su, Y. Wang, D. Cai, S. Baker, A. Korhonen, and N. Collier. PROTOTYPE-TO-STYLE: dialogue generation with style-aware editing on retrieval memory. *IEEE ACM Trans. Audio Speech Lang. Process.*, 29:2152–2161, 2021.
- [36] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. LINE: large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pages 1067–1077. ACM, 2015.
- [37] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018*,. OpenReview.net, 2018.
- [38] N. Wang, L. Lin, J. Li, and H. Wang. Unbiased graph embedding with biased graph observations. In *Proceedings of the ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 1423–1433. ACM, 2022.
- [39] Y. Wang, Y. Zhao, Y. Dong, H. Chen, J. Li, and T. Derr. Improving fairness in graph neural networks via mitigating sensitive attribute leakage. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pages 1938–1948. ACM, 2022.
- [40] H. Wei, H. Huang, T. Zhang, X. Shi, and H. Jin. Enhance temporal knowledge graph completion via time-aware attention graph convolutional network. In *Proceedings of European Conference on Machine Learning and Principles of Knowledge Discovery in Databases, ECML PKDD 2022, Grenoble, France, September 19-23, 2022, Part II*, pages 122–137. Springer, 2022.
- [41] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? In *Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.
- [42] H. Yin, M. Zhang, J. Wang, and P. Li. Surel+: Moving from walks to sets for scalable subgraph-based graph representation learning. In *Proceedings of the 49th International Conference on Very Large Data Bases, VLDB 2023, Vancouver, Canada, August 28 - September 1, 2023*, page 2939–2948, aug 2023.
- [43] S. Yun, S. Kim, J. Lee, J. Kang, and H. J. Kim. Neo-gnns: Neighborhood overlap-aware graph neural networks for link prediction. In *Proceedings of the 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.

- [44] M. Zhang and Y. Chen. Link prediction based on graph neural networks. In *Proceedings of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 5171–5181, 2018.
- [45] 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 *Proceedings of the 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.
- [46] T. Zhao, G. Liu, D. Wang, W. Yu, and M. Jiang. Learning from counterfactual links for link prediction. In *Proceedings of the International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 26911–26926. PMLR, 2022.
- [47] W. X. Zhao, S. Mu, Y. Hou, Z. Lin, Y. Chen, X. Pan, K. Li, Y. Lu, H. Wang, C. Tian, Y. Min, Z. Feng, X. Fan, X. Chen, P. Wang, W. Ji, Y. Li, X. Wang, and J. Wen. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 4653–4664. ACM, 2021.
- [48] W. Zhou, H. Huang, R. Shi, X. Song, X. Lin, X. Wang, and H. Jin. Temporal heterogeneous information network embedding via semantic evolution. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–12, 2023.

## A Further Analysis on the Role of Cross-links

### A.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 [2] 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 [6] and revise its formula to simulate the information propagation with randomly initialized node embeddings:

$$\mathbf{Z}_i^t = \frac{\mathbf{Z}_i^{t-1} + \sum_{j \in \mathcal{N}_i} w_{ij} \mathbf{Z}_j^{t-1}}{|\mathcal{N}_i| + 1} \quad (9)$$

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 [5, 10] to measure the extent of the information cocoons phenomenon in a network, which can be calculated as follows:

$$CH_C = \frac{SSB_C(\mathbf{Z}^t)}{SSW_C(\mathbf{Z}^t)} \cdot \frac{N - C}{C - 1} \quad (10)$$

where  $SSW_C$  and  $SSB_C$  are functions to measure the within-cluster dispersion and between-cluster dispersion, respectively [10].  $N$  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.

• **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.

### A.2 The relationship between cross-links and graph connectivity

By borrowing the concept of network diffusion, we try to explore the role of cross-links in graph connectivity in this part. Specifically, we apply a classic model in network diffusion: the SI model [18], 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<sup>7</sup> as an example. All nodes are divided into four non-overlapping

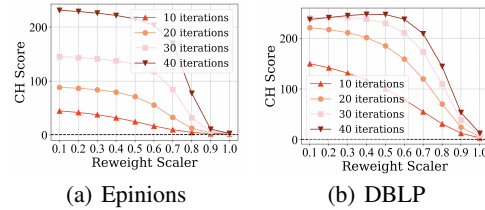


Figure C1: The distribution of CH index score wrt. the weight of cross-links during propagation. A higher value indicates more severe information cocoons. The dashed lines indicate the CH index score under a normal setting.

<sup>7</sup>[https://en.wikipedia.org/wiki/Zachary%27s\\_karate\\_club](https://en.wikipedia.org/wiki/Zachary%27s_karate_club)

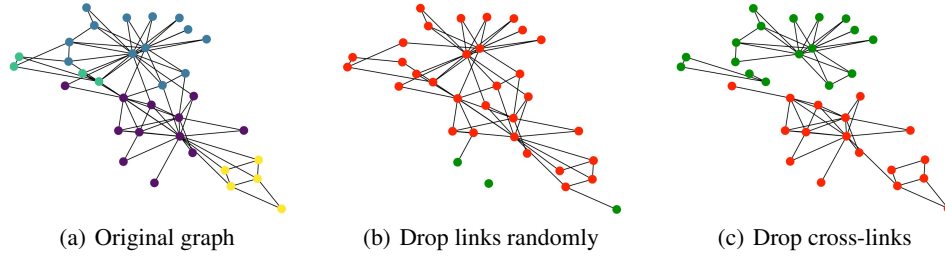


Figure C2: The visualization of diffusion simulation with SI model. Red nodes denote *infected* nodes, and green nodes represent *susceptible* nodes. (a) The original graph of the karate club. Nodes with the same color denote a community. (b) Infected graph after dropping some random edges. (c) Infected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

communities by Louvain algorithm [2] 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 plays a role in bridging two different communities during network diffusion, and it would be hard for a node to send or receive messages from other communities without enough cross-links. In this way, the existence of cross-links plays a key role in preserving graph connectivity.

## B Additional Experimental Settings

### B.1 Base Models and Baselines

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

#### (1) Base models

- **GraphSAGE** [13]: 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** [41]: GIN is a graph neural network that is theoretically as powerful as the Weisfeiler-Lehman test with injective aggregation, combination, and readout functions.
- **GAT** [37]: GAT deploys the attention mechanism during aggregation to capture the neighborhood information with different weights.
- **PPRGo**[3]: 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** [14]: 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** [27]: Based on LightGCN, UltraGCN further theoretically simplifies the model architecture with approximating the infinite-layer information propagation and aggregation.

#### (2) Baselines



- **FairWalk** [30]: Instead of a 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** [4]: 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** [20]: 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** [28]: 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** [38]: 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.

## B.2 Evaluation Metric

In this work, we apply Hits@50, which is widely adopted in other research [43, 45] and OGB leaderboard [15], as our main evaluation metric to measure the link prediction performance of different GNN models. The Hits@50 can be computed by:

$$Hits@50 = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \mathbb{I}(rank_i < 50) \quad (11)$$

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  $i$ th sample.

## B.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<sup>8</sup>. 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 [13], GIN [41], GAT [37], PPRGo [3], LightGCN [14], and UltraGCN [27]. 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),  $\alpha$  is set to be 0.005, and  $T$  is selected from {10, 25, 50} depending on the datasets and base models, and  $\beta$  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  $\alpha$  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,  $\lambda$  as 0.8,  $\gamma$  as 3.5.

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  $\lambda$  to 10; for FLIP, we take the suggestions in the original implementation, and the settings are  $\alpha(0.1)$ ,  $\beta(0.2)$ ; for UGE, we deploy the weighting-based variant as our baseline given that there is no non-sensitive attribute in our settings.

## B.4 Details on LastFM Dataset

Due to the heterogeneity of the recommendation datasets, it is hard to directly deploy community detection algorithms on the original networks and define the corresponding internal-links and cross-links.

<sup>8</sup>Here we follow the evaluation protocol in OGB [15], which is widely used in research.

To this end, inspired by ItemCF [22, 31], we first generate an item-item graph according to the co-occurrence relationship. For example, given a pair of items  $\langle v_1, v_2 \rangle$ , 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, 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:

$$\mathbf{P} = \mathbf{A} \times \mathbf{I}^2 \quad (12)$$

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 compare 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.

## C Further Experiments and Analysis

### C.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.

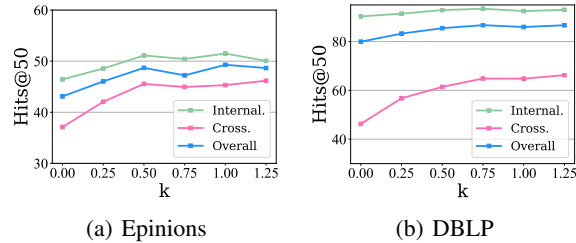


Figure C3: The impact of augmentation size

### C.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 [17] 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.

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.

### C.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 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.

Table C2: Ablation study with METIS community detection algorithm on two real-world datasets (Hits@50)

		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	<b>39.06</b>	<b>28.93</b>	<b>35.89</b>	<b>10.13</b>	<b>78.67</b>	<b>34.65</b>	<b>65.22</b>	<b>44.02</b>
GAT	Orig.	38.33	34.77	37.15	3.56	68.62	28.15	56.26	40.47
	Debias	<b>39.96</b>	<b>36.88</b>	<b>38.86</b>	<b>3.08</b>	<b>75.94</b>	<b>42.77</b>	<b>65.81</b>	<b>33.17</b>
UlltraGCN	Orig.	27.27	11.62	20.77	15.65	97.34	70.57	89.16	26.77
	Debias	<b>46.92</b>	<b>38.18</b>	<b>44.04</b>	<b>8.74</b>	<b>97.47</b>	<b>73.67</b>	<b>90.20</b>	<b>23.80</b>

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