RELIANT: Fair Knowledge Distillation for Graph Neural Networks

Yushun Dong* Binchi Zhang* Yiling Yuan† Na Zou‡ Qi Wang§ Jundong Li*

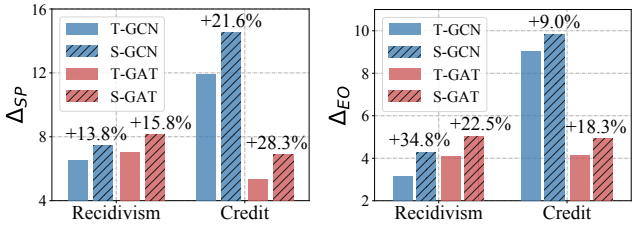
Abstract

Graph Neural Networks (GNNs) have shown satisfying performance on various graph learning tasks. To achieve better fitting capability, most GNNs are with a large number of parameters, which makes these GNNs computationally expensive. Therefore, it is difficult to deploy them onto edge devices with scarce computational resources, e.g., mobile phones and wearable smart devices. Knowledge Distillation (KD) is a common solution to compress GNNs, where a light-weighted model (i.e., the student model) is encouraged to mimic the behavior of a computationally expensive GNN (i.e., the teacher GNN model). Nevertheless, most existing GNN-based KD methods lack fairness consideration. As a consequence, the student model usually inherits and even exaggerates the bias from the teacher GNN. To handle such a problem, we take initial steps towards fair knowledge distillation for GNNs. Specifically, we first formulate a novel problem of fair knowledge distillation for GNN-based teacher-student frameworks. Then we propose a principled framework named RELIANT to mitigate the bias exhibited by the student model. Notably, the design of RELIANT is decoupled from any specific teacher and student model structures, and thus can be easily adapted to various GNN-based KD frameworks. We perform extensive experiments on multiple real-world datasets, which corroborates that RELIANT achieves less biased GNN knowledge distillation while maintaining high prediction utility. Open-source code can be found at https://github.com/yushundong/RELIANT.

Keywords: Graph Neural Networks, Algorithmic Fairness, Knowledge Distillation

1 Introduction

In recent years, Graph Neural Networks (GNNs) have shown satisfying performance in a plethora of real-world applications, e.g., medical diagnosis [27] and credit risk scoring [30], to name a few. In practice, the depth and the number of parameters of GNNs largely determine their expressive power [16], which directly influence their performances in various graph learning tasks [2]. Typically, deeper GNN layers enable the model to capture information that is multiple hops away from any node [21], while a larger number of learnable parameters enable GNN to fit more complex underlying data patterns [2]. However, in most cases, the inference efficiency of GNNs is inevitably degraded by the deep layers or the large number of parameters. Such efficiency degradation naturally makes these GNNs inapplicable to be deployed on edge devices (e.g., mobile phones) with limited computational resources [16, 19].



(a) Bias under Δ_{SP} on CPF. (b) Bias under Δ_{EO} on AKD.

Figure 1: A comparison of exhibited bias between teacher and student models based on two representative GNN knowledge distillation frameworks (CPF and GraphAKD). "T" and "S" represent the teacher and the student model, respectively. The names of GNN mark out the corresponding teacher models.

Due to the problem above, it is necessary to compress those computationally expensive GNNs for deployment on edge devices. Knowledge Distillation (KD) is a common approach to compress GNNs but still maintains a similar level of prediction performance [34, 16, 19]. Here, the basic idea of KD is to let a light-weighted student model (as the compressed GNN) learn to mimic the behavior (e.g., output logits) of the teacher model (usually a computationally expensive GNN). However, most existing KD approaches do not have any fairness consideration over different demographic subgroups, and the optimized student model often preserves and even exaggerates the exhibited bias from the teacher GNN. Consequently, when the compressed model is deployed in real-world application scenarios, there could exist discrimination toward specific populations. Here we provide preliminary analysis based on two representative GNN knowledge distillation frameworks, namely CPF [34] and GraphAKD [16]. Specifically, we measure the exhibited

*University of Virginia, Email: {yd6eb, epb6gw, jundong}@virginia.edu
†Beijing University of Posts and Telecommunications, Email: yuanyiling@bupt.edu.cn
‡Texas A&M University, Email: nzou1@tamu.edu
§Northeastern University, Email: q.wang@northeastern.edu

bias in the widely-studied node classification task on two real-world datasets. Here Recidivism is a network of defendants [18, 1], while Credit is a network between bank clients [35, 1]. We adopt two traditional metrics, i.e., Δ_{SP} (measuring the level of bias under Statistical Parity [10]) and Δ_{EO} (measuring the level of bias under Equal Opportunity [15]), to measure the exhibited bias of GNN predictions. We present a comparison of the exhibited bias between teacher and student models in Fig. 1. Empirical results show that student models tend to yield more biased results compared with the teacher GNN model, which could be attributed to the biased guidance from the teacher GNN during training. It is worth noting that in most cases, directly retraining the teacher GNN for debiasing is infeasible, since retraining the teacher GNN with a large number of parameters is computationally expensive. Hence, mitigating the bias for the student model is an urgent need.

Despite the necessity of mitigating bias for the student model, existing exploration remains scarce. In this paper, we aim to make an initial step towards developing a debiasing framework that can be easily adapted to various existing GNN-based KD methods. However, this task is non-trivial mainly due to the following three challenges: (1) **Gap towards Fair Knowledge:** For most KD frameworks designed for compressing GNNs, the teacher GNN model usually serves as the sole source of supervision signal for the training of the student model. Therefore, if the teacher GNN exhibits any bias, such biased knowledge tends to be inherited by the student model. Hence, learning a fair student model with biased supervision from the teacher GNN is our first challenge. (2) **Gap towards End-to-End Learning:** A critical advantage of existing KD models is the end-to-end learning paradigm, which enables the distilled knowledge to be tailored to specific downstream tasks. In such an end-to-end learning process, highly efficient gradient-based optimization techniques are widely adopted. However, widely-used fairness notions (e.g., Statistical Parity and Equal Opportunity) are defined on the predicted labels. Hence the corresponding bias metrics are naturally non-differentiable w.r.t. the student model parameters. Developing a debiasing framework suitable for gradient-based optimization techniques in the end-to-end learning paradigm is our second challenge. (3) **Gap towards Generalization:** Various KD models have been proposed for compressing GNNs to satisfy different application scenarios. In fact, most KD models are developed based on certain designs of student models. Developing a framework that is student-agnostic and easily adapted to different KD models is our third challenge.

To tackle the above challenges, in this paper, we propose a novel framework named RELIANT (faiR knowlEdge distiLlatIon for grAph Neural neTworks) to mitigate the bias learned by the student model. Specifically, we first formulate a novel research problem of *Fair Knowledge Distillation for GNN-based Teacher-Student Frameworks*. To tackle the first challenge, we incorporate a learnable proxy of the exhibited bias for the student model. In this way, despite the knowledge (from the teacher GNN) being biased, the student model still makes less biased predictions under proper manipulations on the proxy. To tackle the second challenge, we propose to approximate the bias level of the student model, where the approximation is differentiable (w.r.t. the student model parameters) manner. In this way, the highly efficient end-to-end learning paradigm is preserved, and the gradient-based optimization techniques are still applicable. To tackle the third challenge, we design the proposed framework RELIANT in a student-agnostic manner. In other words, the debiasing for the student model does not rely on any specific design tailored for the student model structure. Therefore, RELIANT can be easily adapted to different GNN-based knowledge distillation approaches. The main contributions of this paper are summarized as follows.

- **Problem Formulation.** We formulate and make an initial investigation on a novel research problem of fair knowledge distillation for GNN-Based teacher-student frameworks.
- **Algorithmic Design.** We propose a principled framework named RELIANT that learns the proxy of bias for the student model during KD. RELIANT achieves student-agnostic debiasing via manipulating the proxy during inference.
- **Experimental Evaluation.** We conduct comprehensive experiments on multiple real-world datasets to verify the effectiveness of the proposed framework RELIANT in learning less biased student models.

2 Problem Definition

Notations. We denote matrices, vectors, and scalars by bold uppercase letters (e.g., $\mathbf{X}$), bold lowercase letters (e.g., $\mathbf{x}$), and regular lowercase letters (e.g., x), respectively. For any matrix, e.g., $\mathbf{X}$, we use $\mathbf{X}_{i,j}$ to indicate the element at the i-th row and j-th column.

Preliminaries. We utilize $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{X}\}$ to denote an attributed network (graph). Here, $\mathcal{V} = \{v_1, ..., v_n\}$ is the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, and $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ ($\mathbf{x}_i \in \mathbb{R}^d$, $1 \leq i \leq n$) is the set of node attribute vectors. We use $\mathbf{A} \in \{0, 1\}^{n \times n}$ to denote the adjacency matrix of the graph. If there is an edge from the i-th node to the j-th node, $\mathbf{A}_{i,j} = 1$; otherwise $\mathbf{A}_{i,j} = 0$. Moreover, we denote the pre-trained teacher GNN model in a knowledge distillation framework as $f_{\hat{\boldsymbol{\theta}}}$ parameterized by $\hat{\boldsymbol{\theta}}$. Here $\hat{\boldsymbol{\theta}}$ denotes the optimized $\boldsymbol{\theta}$ of the pre-trained teacher model. Similarly,

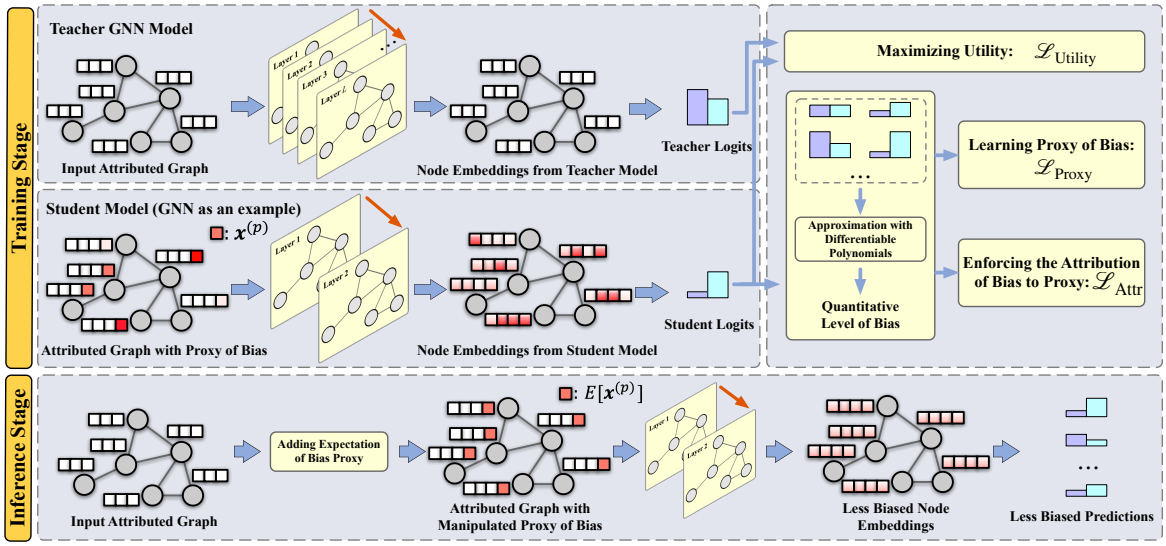


Figure 2: An overview of the proposed framework RELIANT including the training and inference stage.

we denote the student model as $g_{\boldsymbol{\phi}}$ parameterized by $\boldsymbol{\phi}$. We represent the optimized $\boldsymbol{\phi}$ after the training of the student model as $\hat{\boldsymbol{\phi}}$. Without loss of generality, we consider the most widely studied node classification as the downstream task. For the teacher model $f_{\hat{\boldsymbol{\theta}}}(v)$, we denote the set of outcome logits, i.e., the continuous output vector corresponding to each node, as $\hat{\mathcal{Y}}^{(\mathrm{t})} = \{\hat{\mathbf{y}}_1^{(\mathrm{t})}, \hat{\mathbf{y}}_2^{(\mathrm{t})}, ..., \hat{\mathbf{y}}_n^{(\mathrm{t})}\}$, where $\hat{\mathbf{y}}_i^{(\mathrm{t})} \in \mathbb{R}^c$. Here c is the total number of node classes. Correspondingly, we represent the set of outcome logits of the student model $g_{\boldsymbol{\phi}}(v)$ as $\hat{\mathcal{Y}}^{(\mathrm{s})} = \{\hat{\mathbf{y}}_1^{(\mathrm{s})}, \hat{\mathbf{y}}_2^{(\mathrm{s})}, ..., \hat{\mathbf{y}}_n^{(\mathrm{s})}\}$. For any node v_i, the predicted label given by the student model (denoted as $\hat{Y}_i^{(s)}$ for the i-th node) is determined by the largest value across all c dimensions in $\hat{\mathbf{y}}_i^{(\mathrm{s})}$.

Based on the definitions above, we formulate the problem of *Fair Knowledge Distillation for GNN-based Teacher-Student Frameworks* as follows.

PROBLEM 1. ***Fair Knowledge Distillation for GNN-Based Teacher-Student Frameworks.*** *Given an attributed network $\mathcal{G}$ and a GNN-based teacher-student framework including a trained teacher GNN $f_{\hat{\boldsymbol{\theta}}}$ and a student model $g_{\boldsymbol{\phi}}$ to be trained, our goal is to achieve a more fair student model with similar prediction utility compared with $f_{\hat{\boldsymbol{\theta}}}$ for the node classification task.*

3 Methodology

In this section, we first present an overview of the proposed framework RELIANT, followed by the objective function formulation and optimization strategy.

3.1 Workflow of RELIANT

Here we first introduce the workflow of the proposed framework RELIANT. In general, we introduce the three main functionalities involved in the proposed framework RELIANT, namely *maximizing the utility*, *learning proxy of bias*, and *enforcing the attribution of bias to the proxy*. We present an overview of RELIANT in Fig. 2. Specifically, to tackle the first challenge (gap towards fair knowledge), we propose to first learn the proxy of bias as extra input attributes for the student model to account for the exhibited bias (on training nodes), and then exclude the information of proxy during test with manipulated pseudo proxy. To tackle the second challenge (gap towards end-to-end learning), we formulate our debiasing objectives in a differentiable (w.r.t. the parameters of the student model) manner. To tackle the third challenge (gap towards generalization), we achieve debiasing in a student-agnostic manner. In other words, the proposed framework RELIANT does not rely on any specific student model structure to achieve debiasing. We elaborate more details as follows.

Maximizing Utility. In general, existing GNN-based KD frameworks consider the GNNs with high computational costs as the teacher model, and the goal is to learn a student model with limited computational costs but similar prediction utility (e.g., accuracy in node classification tasks). To maintain the utility of the teacher model, it is necessary to utilize the knowledge from the teacher model as the supervision signal for the training of the student. In particular, a common approach is to utilize the output classification logits from the teacher model as the supervision signal, which we take as an example here. Specifically, we minimize the distance between the logits from the student model and the teacher model. We formally formulate the optimization goal as

$$\min_{\boldsymbol{\phi}} \sum_{v_i \in \mathcal{V}} \gamma_{\mathrm{d}} \left(\hat{\mathbf{y}}_i^{(t)}, \hat{\mathbf{y}}_i^{(s)} \right), \tag{3.1}$$

where $\hat{\mathbf{y}}_i^{(s)}$ and $\hat{\mathbf{y}}_i^{(t)}$ are the output logits from the student model $g_{\boldsymbol{\phi}}(v_i)$ and teacher model $f_{\hat{\boldsymbol{\theta}}}(v_i)$, respectively. The function $\gamma_{\mathrm{d}}(.,.)$ measures the distance between two logit vectors. Different choices can be adopted to measure the distance, e.g., cosine distance and Euclidean distance. Correspondingly, to maximize the prediction utility, we minimize the objective function

$$\mathscr{L}_{\mathrm{Utility}}(\boldsymbol{\phi}) = \sum_{v_i \in \mathcal{V}} \gamma_{\mathrm{d}} \left(\hat{\mathbf{y}}_i^{(t)}, \hat{\mathbf{y}}_i^{(s)} \right). \tag{3.2}$$

Learning Proxy of Bias. It is worth noting that even if the sensitive attributes are removed from the input data, the student model could still exhibit bias in its predictions. The main reason is that there could exist dependencies between those sensitive attributes and non-sensitive ones. Moreover, the information about sensitive attributes could also be encoded in the input network structure [6]. As a consequence, it is difficult to prevent the student model from leveraging information about sensitive attributes. To handle such a problem, we propose to learn the proxy of bias $\mathbf{x}_i^{(p)}$ as extra input attributes for each node v_i. Here, the rationale is that if much information about bias comes from the learned proxy instead of those encoded in the non-sensitive attributes or the network structures, then we are able to achieve less biased predictions by not using the information from such a proxy during test. As a consequence, such a proxy of bias should account for the exhibited bias of the student model as much as possible. In other words, the exhibited bias should be largely attributed to the proxy of bias rather than the sensitive information encoded in the network data. More specifically, to enforce the proxy of bias contributing to the exhibited bias in the student model, we propose to maximize the exhibited bias when these proxies are taken as input into the student model together with other attributes and the network structure. We formally formulate our goal as

$$\max_{\mathbf{X}^{(p)}} J_{\mathrm{Bias}}(\{g_{\boldsymbol{\phi}}(\gamma(v_i, \mathbf{X}^{(p)})) : i \in \mathcal{V}\}), \tag{3.3}$$

where $\gamma(.,.)$ is a function that takes a node and the proxy of bias matrix as input, and outputs the node with a concatenated node attribute vector $[\mathbf{x}_i, \mathbf{x}_i^{(p)}]$. Here $\mathbf{x}_i^{(p)}$ is the i-th row of $\mathbf{X}^{(p)}$. $J_{\mathrm{Bias}}(.)$ is a function that takes the set of logits from the student model as input and outputs a value indicating the level of exhibited bias. Nevertheless, the computation is non-differentiable under traditional fairness notions such as Statistical Parity and Equal Opportunity. Here we propose to utilize orthogonal polynomials (e.g., Legendre polynomials [4]) that are differentiable w.r.t. the output logits to approximate the level of bias under traditional fairness notions. This makes J_{Bias} differentiable w.r.t. the learnable parameter $\boldsymbol{\phi}$. Correspondingly, we formally give the objective function towards the goal above as

$$\mathscr{L}_{\mathrm{Proxy}}(\mathbf{X}^{(p)}) = -J_{\mathrm{Bias}}(\{g_{\boldsymbol{\phi}}(\gamma(v_i, \mathbf{X}^{(p)})) : i \in \mathcal{V}\}). \tag{3.4}$$

Enforcing the Attribution of Bias to the Proxy. Only achieving Eq. (3.3) is not enough to enforce the proxy of bias largely accounting for the exhibited bias of the student model. This is because the vanilla node attributes could still contribute to the exhibited bias. More specifically, we denote $P(\hat{Y}^{(s)})$ as the probability of a certain classification prediction given by the student model for any specific node. We assume that there are underlying unbiased and biased node attributes $\mathbf{X}^{(u)}$ and $\mathbf{X}^{(b)}$, respectively. When Eq. (3.3) is achieved, it is clear that $P(\hat{Y}^{(s)}|\mathbf{X}^{(u)}, \mathbf{X}^{(b)}, \mathbf{X}^{(p)})$, i.e., $P(\hat{Y}^{(s)}|\mathbf{X}, \mathbf{X}^{(p)})$, is biased. However, both $\mathbf{X}^{(b)}$ and $\mathbf{X}^{(p)}$ could be the source of the exhibited bias. It is worth noting that our goal is to learn proxy $\mathbf{X}^{(p)}$ to account for as much of the exhibited bias as possible. Therefore, to enforce the effectiveness of the proxy, it is necessary to ensure that the exhibited bias is attributed to the biased information from $\mathbf{X}^{(p)}$ instead of $\mathbf{X}^{(b)}$. In other words, we need to enforce $P(\hat{Y}^{(s)}|\mathbf{X}^{(u)}, \mathbf{X}^{(b)})$ being less biased, which ensures that $\mathbf{X}^{(p)}$ accounts for the exhibited bias as much as possible. Nevertheless, $P(\hat{Y}^{(s)}|\mathbf{X}^{(u)}, \mathbf{X}^{(b)})$ is intractable considering that the number of the input dimension number for the student model is fixed. Hence we propose an alternative approach. Denote the learned proxy of bias and the underlying sensitive attribute vector of any node as $\mathbf{x}^{(p)}$ and $\mathbf{s}$, respectively. We propose to utilize a vector $\mathbb{E}[\mathbf{x}^{(p)}]$ to replace each row in $\mathbf{X}^{(p)}$ as the manipulated pseudo proxy $\tilde{\mathbf{X}}^{(p)}$. In this way, the rows in $\tilde{\mathbf{X}}^{(p)}$ are independent from $\mathbf{s}$, i.e., the information about sensitive attributes encoded in $\mathbf{X}^{(p)}$ is wiped out. To enforce the attribution of bias to the proxy $\mathbf{X}^{(p)}$, the predictions should be as fair as possible when the information about $\mathbf{X}^{(p)}$ is removed. Therefore, we formulate our last optimization goal as

$$\min_{\boldsymbol{\phi}} J_{\mathrm{Bias}}(\{g_{\boldsymbol{\phi}}(\tilde{\gamma}(v_i, \tilde{\mathbf{X}}^{(p)})) : i \in \mathcal{V}\}), \tag{3.5}$$

where $\tilde{\gamma}(.,.)$ is a function that takes a node and the matrix $\tilde{\mathbf{X}}^{(p)}$ as input, and returns the input node with a concatenated node attribute vector $[\mathbf{x}_i, \tilde{\mathbf{x}}_i^{(p)}]$. Here $\tilde{\mathbf{x}}_i^{(p)}$ is the i-th row of matrix $\tilde{\mathbf{X}}^{(p)}$. We formally present the corresponding objective function as

$$\mathscr{L}_{\mathrm{Attr}}(\boldsymbol{\phi}) = J_{\mathrm{Bias}}(\{g_{\boldsymbol{\phi}}(\tilde{\gamma}(v_i, \tilde{\mathbf{X}}^{(p)})) : i \in \mathcal{V}\}). \tag{3.6}$$

Inference with Student Model. To achieve less biased inference, an ideal case is to make predictions

with $P(\hat{Y}^{(s)}|\mathbf{X}^{(u)})$. However, it is difficult to explicitly extract $\mathbf{X}^{(u)}$ from $\mathbf{X}$. Instead, we argue that $P(\hat{Y}^{(s)}|\mathbf{X}^{(u)}, \mathbf{X}^{(b)}, \tilde{\mathbf{X}}^{(p)})$ exhibits similar level of bias compared with $P(\hat{Y}^{(s)}|\mathbf{X}^{(u)})$. This is because (1) the bias exhibited by $P(\hat{Y}^{(s)}|\mathbf{X}^{(u)}, \mathbf{X}^{(b)}, \tilde{\mathbf{X}}^{(p)})$ minimally relies on $\mathbf{X}^{(b)}$ after enforcing Eq. (3.3) and Eq. (3.5); and (2) there is no further information about sensitive attributes encoded in $\tilde{\mathbf{X}}^{(p)}$ (as discussed above). Consequently, we propose to utilize $g_{\boldsymbol{\phi}}(\tilde{\gamma}(v_i, \tilde{\mathbf{X}}^{(p)}))$ to achieve less biased prediction for node v_i in the inference stage.

4 Optimization Objectives & Strategy

We present the optimization objectives of RELIANT followed by the training strategy in this section.

Optimization Objectives. Based on our discussions above, here we present a summary of the optimization objectives for the proposed RELIANT. First, to optimize the parameter $\boldsymbol{\phi}$, we formally formulate a unified objective function as

$$\mathscr{L}_{\boldsymbol{\phi}} = \mathscr{L}_{\text{Utility}}(\boldsymbol{\phi}) + \lambda \cdot \mathscr{L}_{\text{Attr}}(\boldsymbol{\phi}). \tag{4.7}$$

Here λ serves as a hyper-parameter controlling the effect of debiasing the student model. Second, to optimize the learnable proxy of bias $\mathbf{X}^{(p)}$, we formally present the objective function as

$$\mathscr{L}_{\mathbf{X}^{(\text{P})}} = \mathscr{L}_{\text{Proxy}}(\mathbf{X}^{(p)}). \tag{4.8}$$

Optimization Strategy. To train the proposed framework RELIANT, we propose to optimize the parameter $\boldsymbol{\phi}$ and learnable proxy of bias $\mathbf{X}^{(p)}$ in an alternating manner. We present the algorithmic routine of RELIANT in Algorithm 1.

Algorithm 1 Fair Knowledge Distillation for GNNs

Input: $\mathcal{G}$: the graph data; $f_{\hat{\boldsymbol{\theta}}}$: the trained teacher GNN model; $g_{\boldsymbol{\phi}}$: the student model;
Output: $g_{\hat{\boldsymbol{\phi}}}$: the optimized student model; $\mathbf{X}^{(p)}$: the proxy of bias matrix;
1: Randomly initialize $\mathbf{X}^{(p)}$;
2: **while** stop training condition not satisfied **do**
3: Compute $\mathscr{L}_{\boldsymbol{\phi}}$ according to Eq. (4.7);
4: Update $\boldsymbol{\phi}$ with $\frac{\partial \mathscr{L}_{\boldsymbol{\phi}}}{\partial \boldsymbol{\phi}}$;
5: Compute $\mathscr{L}_{\mathbf{X}^{(p)}}$ according to Eq. (4.8);
6: Update $\mathbf{X}^{(p)}$ with $\frac{\partial \mathscr{L}_{\mathbf{X}^{(p)}}}{\partial \mathbf{X}^{(p)}}$;
7: **end while**
8: **return** $g_{\hat{\boldsymbol{\phi}}}$ and $\mathbf{X}^{(p)}$;

5 Experimental Evaluations

In this section, we will first introduce the downstream learning task and adopted real-world datasets, followed by the backbone models, baseline methods, and evaluation metrics. Next, we present the implementation details of the models. Finally, we discuss the evaluation results of the proposed RELIANT. In particular, we aim to answer the following research questions through experiments: **RQ1:** How well can RELIANT balance the utility and fairness of the student model compared with other baselines? **RQ2:** To what extent each component of RELIANT contributes to the overall debiasing performance? **RQ3:** How will the choice of the hyper-parameter λ affect the performance of RELIANT?

5.1 Experimental Settings

Here we introduce the settings for our experimental evaluation.

Downstream Task & Real-world Datasets. We adopt the widely studied node classification as the downstream task in this paper. We adopt four real-world datasets for the experimental evaluations, including two widely used network datasets (Recidivism [18, 1] and Credit Defaulter [35, 1]) and two newly constructed ones based on real-world data (DBLP and DBLP-L). In Recidivism, nodes are defendants released on bail, and edges denote the connections between defendants computed from their past criminal records. Here the sensitive feature is race, and we aim to classify if a certain defendant is unlikely to commit a crime after bail. In Credit Defaulter, nodes are credit card users, and edges are the connections between these users. Here we consider the age period of these users as their sensitive feature, and we aim to predict the future default of credit card payments. Additionally, we also construct two co-author networks, namely DBLP and DBLP-L based on AMiner network [29], which is a co-author network collected from computer science bibliography. Specifically, we first filter out the nodes in AMiner network with incomplete information. Then we adopt two different approaches to sample a connected network from the filtered dataset: DBLP is a subgraph sampled with random walk, while DBLP-L is the largest connected component of the filtered AMiner network. In both datasets, nodes represent the researchers in different fields, and edges denote the co-authorship between researchers. The sensitive attribute is the continent of the affiliation each researcher belongs to, and we aim to predict the primary research field of each researcher. The detailed statistics of these four datasets are in Table 1.

KD Framework Backbones & Teacher GNNs. To evaluate the capability of RELIANT in generalizing to different GNN-based KD backbones, here we adopt two representative KD frameworks designed for compressing GNNs, namely CPF [34] and GraphAKD [16]. In general, CPF minimizes the distribution distance between the logits from teacher and student to provide supervision information for the student, while GraphAKD utilizes adversarial training to achieve knowledge distillation for the student. The student model of CPF and GraphAKD is PLP [34] and SGC [32], respectively. For each KD

Table 1: The basic information about the real-world datasets adopted for experimental evaluation. Sens. denotes the semantic meaning of sensitive attribute.

Dataset	**Recidivism**	**Credit Defaulter**	**DBLP**	**DBLP-L**
# Nodes	18,876	30,000	39,424	129,726
# Edges	321,308	1,436,858	52,460	591,039
# Attributes	18	13	5,693	5,693
Avg. degree	34.0	95.8	1.3	4.6
Sens.	Race	Age	Continent of Affiliation	Continent of Affiliation
Label	Bail Decision	Future Default	Research Field	Research Field

Table 2: The experimental results based on node classification accuracy and Δ_{SP}. We use "(T)" and "(S)" suffixes to represent the teacher model and the student model, respectively. Here Vanilla(S) denotes the student model trained with the vanilla KD framework; One-Hot(S) represents the student model trained with the one-hot bias proxy; RELIANT(S) is the student model trained with our proposed model. ↑ denotes the larger, the better; while ↓ denotes the opposite. All quantitative results are presented in percentages. The best results are in **Bold**.

			DBLP	**DBLP-L**	**Credit**	**Recidivism**
CPF +GCN	**Accuracy (↑)**	**GCN(T)**	92.37 ± 0.06	94.20 ± 0.09	76.39 ± 0.48	**93.68 ± 0.21**
		Vanilla(S)	**93.14 ± 0.10**	**94.30 ± 0.04**	**77.85 ± 0.10**	89.41 ± 0.12
		One-Hot(S)	93.04 ± 0.34	94.16 ± 0.02	77.65 ± 0.10	89.15 ± 0.37
		RELIANT(S)	92.70 ± 0.40	94.07 ± 0.18	77.82 ± 0.45	88.88 ± 0.57
	Δ_{SP} (↓)	**GCN(T)**	7.66 ± 0.26	7.33 ±0.44	15.81 ±0.40	6.10 ±0.05
		Vanilla(S)	8.55 ± 0.50	7.16 ± 0.16	14.90 ± 0.89	6.85 ± 0.05
		One-Hot(S)	7.97 ± 0.63	7.46 ± 0.24	13.80 ± 0.32	6.78 ± 0.51
		RELIANT(S)	**2.27 ± 1.00**	**3.09 ± 0.36**	**10.28 ± 1.86**	**4.06 ± 0.64**
CPF +SAGE	**Accuracy (↑)**	**SAGE(T)**	92.57 ± 0.28	94.10 ± 0.25	77.88 ± 0.06	**89.71 ± 0.14**
		Vanilla(S)	**93.25 ± 0.15**	**94.97 ± 0.10**	77.97 ± 0.26	89.20 ± 0.11
		One-Hot(S)	93.07 ± 0.10	94.32 ± 0.07	78.01 ± 0.23	89.11 ± 0.29
		RELIANT(S)	92.91 ± 0.51	94.17 ± 0.93	**78.28 ± 0.36**	88.85 ± 0.27
	Δ_{SP} (↓)	**SAGE(T)**	8.32 ±0.24	7.81 ±0.08	14.08 ± 1.37	6.50 ±0.39
		Vanilla(S)	8.29 ± 0.85	7.02 ± 0.13	13.44 ± 5.23	4.41 ± 0.43
		One-Hot(S)	8.01 ± 0.25	7.52 ± 0.32	16.86 ± 3.86	6.62 ± 0.38
		RELIANT(S)	**2.01 ± 1.21**	**2.97 ± 0.61**	**10.06 ± 1.70**	**3.94 ± 0.60**
AKD +GCN	**Accuracy (↑)**	**GCN(T)**	**92.37 ± 0.06**	**94.20 ± 0.09**	**76.39 ± 0.48**	**93.68 ± 0.21**
		Vanilla(S)	92.06 ± 0.16	94.07 ± 0.11	76.35 ± 0.31	92.08 ± 0.29
		One-Hot(S)	91.55 ± 0.40	94.07 ± 0.04	75.65 ± 0.75	92.07 ± 0.03
		RELIANT(S)	91.39 ± 0.24	93.98 ± 0.08	75.64 ± 0.06	91.21 ± 0.14
	Δ_{SP} (↓)	**GCN(T)**	7.66 ±0.26	7.33 ±0.44	15.81±0.40	6.10 ±0.05
		Vanilla(S)	7.87 ± 0.25	6.79 ± 0.10	13.61 ± 2.00	6.54 ± 0.17
		One-Hot(S)	7.39 ± 0.35	6.72 ± 0.19	14.30 ± 0.24	6.44 ± 0.32
		RELIANT(S)	**3.66 ± 1.09**	**5.18 ± 0.16**	**8.47 ± 1.92**	**5.70 ± 0.18**
AKD +SAGE	**Accuracy (↑)**	**SAGE(T)**	**92.57 ± 0.28**	94.10 ± 0.25	77.88 ± 0.06	**89.71 ± 0.14**
		Vanilla(S)	92.23 ± 0.07	94.45 ± 0.03	78.10 ± 0.24	89.67 ± 0.07
		One-Hot(S)	92.31 ± 0.06	**94.52 ± 0.11**	78.24 ± 0.45	89.60 ± 0.12
		RELIANT(S)	92.07 ± 0.07	94.28 ± 0.06	**78.60 ± 0.33**	88.87 ± 0.31
	Δ_{SP} (↓)	**SAGE(T)**	8.32 ±0.24	7.81 ±0.08	14.08 ± 1.37	6.50 ±0.39
		Vanilla(S)	7.53 ± 0.29	7.34 ± 0.41	14.41 ± 0.15	6.24 ± 0.20
		One-Hot(S)	7.72 ± 0.44	7.26 ± 0.36	11.69 ± 0.93	6.18 ± 0.30
		RELIANT(S)	**4.91 ± 0.64**	**4.05 ± 0.14**	**5.00 ± 1.63**	**6.06 ± 0.26**

framework, we adopt two types of GNNs (including GCN [21] and GraphSAGE [14]) as the teacher GNN.

Baselines. To the best of our knowledge, this is the first study on how to mitigate the bias exhibited in GNN-based KD frameworks. In experiments, we adopt the student model yielded by the vanilla KD framework as our first baseline. For our second baseline, we replace the learnable proxy of bias with a naive proxy for the input of the KD framework. Specifically, we utilize one-hot vectors as the naive proxy for different demographic subgroups during training, where the one-hot vector flags the membership of different nodes. We replace all proxy vectors during inference with an averaged proxy vector across all instances. Here, the rationale is that more distinguishable attributes are easier for deep learning models to learn during training, and these one-hot vectors serve as an "easier" indicator of biased information. In this way, if these one-hot proxy accounts for the exhibited bias of the student model after training, then the exhibited bias could also be mitigated during inference, where such information is wiped out.

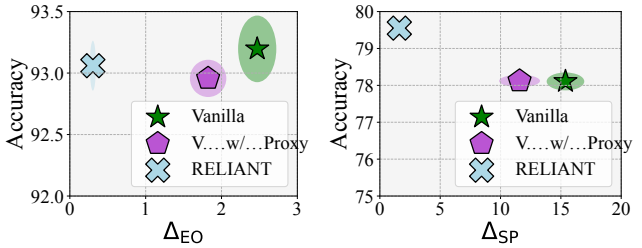
Evaluation Metrics. We evaluate the performance of the compressed GNN models (i.e., the output student model of KD frameworks) from two perspectives, namely

utility and fairness. Specifically, in terms of utility, we adopt the node classification accuracy as the corresponding metric; in terms of fairness, we adopt two traditional metrics Δ_{SP} and Δ_{EO}. Here Δ_{SP} measures the bias level (of predictions) under the fairness notion of Statistical Parity, while Δ_{EO} measures the bias level under the notion of Equal Opportunity. See online version of this paper for other results in Appendix due to space limit.

Implementation Details. RELIANT is implemented in PyTorch [25] and optimized with Adam optimizer [20]. In our experiments, the learning rate is chosen in $\{10^{-2}, 10^{-3}, 10^{-4}\}$ and the training epoch number is set as 1,000 for CPF and 600 for GraphAKD. Experiments are carried out on an Nvidia RTX A6000, and the reported numerical results are averaged across three different runs. We introduce more details in Appendix.

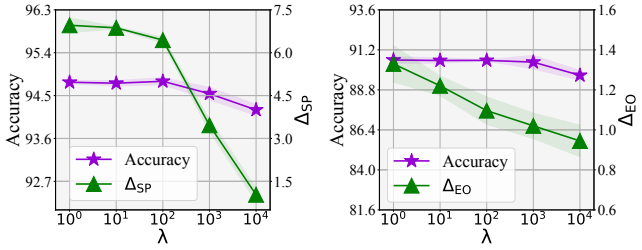
5.2 Effectiveness of RELIANT Here we aim to answer **RQ1**. Specifically, we evaluate our proposed framework RELIANT on two KD backbones, namely CPF and GraphAKD. For each KD backbone, we adopt two different GNNs (GCN and GraphSAGE) to evaluate the capability of our proposed framework in generalizing to different GNNs. We compare the corresponding performances between the teacher GNN model and the student models trained with three different frameworks (i.e., the vanilla KD framework, the KD framework with the one-hot proxy of bias, and our proposed RELIANT). We present quantitative results on node classification accuracy and Δ_{SP} in Table 2. In addition, we also perform experiments based on Equal Opportunity (see Appendix), where we have consistent observations. We make the following observations from Table 2.

- From the perspective of prediction utility, student models trained with all three KD frameworks achieve comparable performances with the teacher model. This implies that effective knowledge distillation can be achieved by all three KD frameworks.
- From the perspective of bias mitigation, the student models trained with the vanilla KD frameworks inherit and even exaggerate the exhibited bias from the teacher GNN model in all cases. Training the student models with the one-hot proxy can mitigate bias in most cases. Compared with the student models trained with the vanilla KD framework and the one-hot proxy, RELIANT consistently exhibits less bias w.r.t. Statistical Parity.
- Based on the performance of RELIANT in both perspectives, RELIANT achieves effective debiasing for the student model but still maintains comparable model utility with the teacher model. Therefore, we argue that RELIANT achieves a satisfying balance between debiasing and maintaining utility.



(a) GraphAKD on Credit. (b) CPF on DBLP.

Figure 3: Ablation study of RELIANT. "Vanilla" denotes the student model trained with the original KD framework, while "V. w/ Proxy" represents the student model trained under the KD framework with only learning the proxy of bias.



(a) CPF on DBLP-L. (b) GraphAKD on Recidivism.

Figure 4: Parameter sensitivity of λ based on two different KD backbones on two real-world datasets. We also have similar observations on other datasets.

5.3 Ablation Study We aim to answer **RQ2** in this subsection. Specifically, for each framework, we evaluate to what extent the two modules of RELIANT (including *learning proxy of bias* and *enforcing the attribution of bias to the proxy*) contribute to the performance of the student model. We present the results in Fig. 3. Here, Fig. 3a is the performance of accuracy vs. Δ_{SP} from CPF based on the DBLP-L dataset, while Fig. 3b is the performance of accuracy vs. Δ_{EO} from GraphAKD based on the Recidivism dataset. Notably, we also have similar observations under other settings. We make the following observations.

- From the perspective of prediction utility, we observe that the prediction utility is comparable among all three cases. This corroborates that both modules exert limited influence on the node classification accuracy.
- From the perspective of bias mitigation, adding the module of *learning proxy of bias* to the vanilla KD framework brings limited bias mitigation. This is because the bias could also come from the non-sensitive node attributes (as discussed in Section 3.1). After the module of *enforcing the attribution of bias to the proxy* is added together with *learning proxy of bias*, RELIANT is then able to achieve satisfying performance on bias mitigation.

5.4 Parameter Sensitivity We answer **RQ3** by studying the tendency of model utility and exhibited bias w.r.t. the change of hyper-parameter λ. Here λ controls the effect of $\mathscr{L}_{\text{Attr}}$. More specifically, we vary λ in $\{10^0, 10^1, 10^2, 10^3, 10^4\}$, and we present the corresponding tendency of node classification accuracy and the exhibited bias of the trained student model with RELIANT in Fig. 4. Here, Fig. 4a is based on the Credit dataset under GraphAKD, while Fig. 4b is based on the DBLP dataset under CPF. We also have similar observations on other datasets. We make the following observations from Fig. 4.

- From the perspective of prediction utility, the node classification accuracies on both datasets and KD backbones do not exhibit apparent reduction when the value of λ increases from 1 to 10^4. This verifies that the prediction utility is not sensitive to λ.
- From the perspective of bias mitigation, the student model exhibits less bias when λ increases from 1 to 10^4. Specifically, when λ is relatively small (e.g., 1), the learned proxy of bias only partially accounts for the exhibited bias; when the value of λ increases, more bias is then attributed to the learned proxy. Considering the balance between model utility and bias mitigation, a recommended range of λ is between 10^2 and 10^3.

6 Related Works

Algorithmic Fairness in GNNs. Most existing works promoting the algorithmic fairness of GNNs focus either on *Group Fairness* [10] or *Individual Fairness* [36]. Specifically, group fairness is defined based on a set of pre-defined sensitive attributes (e.g., gender and race). These sensitive attributes divide the whole population into different demographic subgroups. Group fairness requires that each subgroup should receive their fair share of interest according to the output GNN predictions [23]. Various explorations have been made towards achieving a higher level of group fairness for GNNs [7]. Decoupling the output predictions from sensitive attributes via adversarial learning is one of the most popular approaches among existing works [31, 3]. Other common strategies include reformulating the objective function with fairness regularization [11, 24], rebalancing the number of intra-group edges between two demographic subgroups [6, 22], deleting nodes or edges that contribute the most to the exhibited bias [8, 9], etc. On the other hand, individual fairness does not rely on any sensitive attributes. Instead, individual fairness requires that similar nodes (in the input space) should be treated similarly (in the output space) [10]. To fulfill individual fairness in GNNs, adding fairness-aware regularization terms to the optimization objective is the most widely adopted approach [5, 28].

Knowledge Distillation. In recent years, knowledge distillation has been proven to be effective in compressing the model but still maintaining similar model prediction performance [12]. Correspondingly, it has been widely adopted in a plethora of applications, including visual recognition [33], natural language processing [13, 17], etc. The main idea of knowledge distillation is to transfer the knowledge of a computationally expensive teacher model to a light student model, and thus the student model is able to fit in platforms with limited computing resources [16, 19]. It is worth noting that such a strategy is also proved to be effective in compressing GNNs [34, 16, 19]. Consequently, there is growing research attention on utilizing knowledge distillation to compress GNNs for more efficient inference. For example, encouraging the student model to yield similar output to the teacher GNN via regularization is proved to be effective [16]. In addition, adversarial learning is also a popular technique to obtain light-weighted but accurate student models [16]. However, most of these frameworks for GNNs do not have fairness consideration. Hence the student model tends to be influenced by biased knowledge from the teacher GNN. Different from existing works, we develop a generalizable knowledge distillation framework that explicitly considers fairness in GNNs but still maintains the utility of GNN predictions.

7 Conclusion

Despite the success of Knowledge Distillation (KD) in compressing GNNs, most existing works do not consider fairness. Hence the student model trained with the KD framework tends to inherit and even exaggerate the bias from the teacher GNN. In this paper, we take initial steps towards learning less biased student models for GNN-based KD frameworks. Specifically, we first formulate a novel problem of fair knowledge distillation for GNN-based teacher-student frameworks, then propose a framework named RELIANT to achieve a less biased student model. Notably, the design of RELIANT is agnostic to the specific structures of teacher and student models. Therefore, it can be easily adapted to different KD approaches for debiasing. Extensive experiments demonstrate the effectiveness of RELIANT in fulfilling fairness for GNN compression with KD.

8 Acknowledgments

This work is supported by the National Science Foundation under grants IIS-2006844, IIS-2144209, IIS-2223768, IIS-2223769, CNS-2154962, CMMI-2125326, BCS-2228533, and BCS-2228534, the JP Morgan Chase Faculty Research Award, and the Cisco Faculty Research Award. We would like to thank the anonymous reviewers for their constructive feedback.

References

[1] Agarwal, C., Lakkaraju, H., and Zitnik, M. Towards a unified framework for fair and stable graph representation learning. In *UAI* (2021).

[2] Chen, M., Wei, Z., Huang, Z., Ding, B., and Li, Y. Simple and deep graph convolutional networks. In *ICML* (2020).

[3] Dai, E., and Wang, S. Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information. In *WSDM* (2021).

[4] Dattoli, G., Ricci, P. E., and Cesarano, C. A note on legendre polynomials. *International Journal of Nonlinear Sciences and Numerical Simulation 2*, 4 (2001), 365–370.

[5] Dong, Y., Kang, J., Tong, H., and Li, J. Individual fairness for graph neural networks: A ranking based approach. In *SIGKDD* (2021).

[6] Dong, Y., Liu, N., Jalaian, B., and Li, J. EDITS: modeling and mitigating data bias for graph neural networks. In *WWW* (2022).

[7] Dong, Y., Ma, J., Chen, C., and Li, J. Fairness in graph mining: A survey. *arXiv preprint arXiv:2204.09888* (2022).

[8] Dong, Y., Wang, S., Ma, J., Liu, N., and Li, J. Interpreting unfairness in graph neural networks via training node attribution. In *AAAI* (2023).

[9] Dong, Y., Wang, S., Wang, Y., Derr, T., and Li, J. On structural explanation of bias in graph neural networks. In *SIGKDD* (2022).

[10] Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. Fairness through awareness. In *ITCS* (2012).

[11] Fan, W., Liu, K., Xie, R., Liu, H., Xiong, H., and Fu, Y. Fair graph auto-encoder for unbiased graph representations with wasserstein distance. In *ICDM* (2021).

[12] Gou, J., Yu, B., Maybank, S. J., and Tao, D. Knowledge distillation: A survey. *International Journal of Computer Vision 129*, 6 (2021), 1789–1819.

[13] Haidar, M., Rezagholizadeh, M., et al. Textkd-gan: Text generation using knowledge distillation and generative adversarial networks. In *Canadian conference on artificial intelligence* (2019), Springer, pp. 107–118.

[14] Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *NeurIPS* (2017).

[15] Hardt, M., Price, E., and Srebro, N. Equality of opportunity in supervised learning. In *NeurIPS* (2016).

[16] He, H., Wang, J., Zhang, Z., and Wu, F. Compressing deep graph neural networks via adversarial knowledge distillation. *arXiv preprint arXiv:2205.11678* (2022).

[17] Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., and Liu, Q. Tinybert: Distilling BERT for natural language understanding. In *EMNLP* (2020).

[18] Jordan, K. L., and Freiburger, T. L. The effect of race/ethnicity on sentencing: Examining sentence type, jail length, and prison length. *J Crim Justice* (2015).

[19] Joshi, C. K., Liu, F., Xun, X., Lin, J., and Foo, C.-S. On representation knowledge distillation for graph neural networks. *arXiv preprint arXiv:2111.04964* (2021).

[20] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization. In *ICLR* (2015).

[21] Kipf, T. N., and Welling, M. Semi-supervised classification with graph convolutional networks. In *ICLR* (2017).

[22] Li, P., Wang, Y., Zhao, H., Hong, P., and Liu, H. On dyadic fairness: Exploring and mitigating bias in graph connections. In *ICLR* (2021).

[23] M., N., M., F., S., N., L., K., and G., A. A survey on bias and fairness in machine learning. *CSUR* (2021).

[24] Navarin, N., Oneto, L., and Donini, M. Learning deep fair graph neural networks. In *ECML* (2020).

[25] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. In *NeurIPS* (2017).

[26] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS* (2019).

[27] Saha, P., Mukherjee, D., Singh, P. K., Ahmadian, A., Ferrara, M., and Sarkar, R. Graphcovidnet: A graph neural network based model for detecting covid-19 from ct scans and x-rays of chest. *Scientific Reports 11*, 1 (2021), 1–16.

[28] Song, W., Dong, Y., Liu, N., and Li, J. Guide: Group equality informed individual fairness in graph neural networks. In *KDD* (2022).

[29] Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., and Su, Z. Arnetminer: extraction and mining of academic social networks. In *KDD* (2008).

[30] Wang, D., Zhang, Z., Zhou, J., Cui, P., Fang, J., Jia, Q., Fang, Y., and Qi, Y. Temporal-aware graph neural network for credit risk prediction. In *SDM* (2021).

[31] Wang, Y., Zhao, Y., Dong, Y., Chen, H., Li, J., and Derr, T. Improving fairness in graph neural networks via mitigating sensitive attribute leakage. In *SIGKDD* (2022).

[32] Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., and Weinberger, K. Simplifying graph convolutional networks. In *ICML* (2019).

[33] Wu, X., He, R., Hu, Y., and Sun, Z. Learning an evolutionary embedding via massive knowledge distillation. *International Journal of Computer Vision* (2020).

[34] Yang, C., Liu, J., and Shi, C. Extract the knowledge of graph neural networks and go beyond it: An effective knowledge distillation framework. In *WWW* (2021).

[35] Yeh, I.-C., and Lien, C.-h. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert systems with applications* (2009).

[36] Zemel, R., Wu, Y., Swersky, K., Pitassi, T., and Dwork, C. Learning fair representations. In *ICML* (2013).