Are Your Reviewers Being Treated Equally? Discovering Subgroup Structures to Improve Fairness in Spam Detection

Jiaxin Liu
Lehigh University
Pennsylvania, USA
jilb17@lehigh.edu

Yuefei Lyu
BUPT
Beijing, China
lvyuefei@bupt.edu.cn

Xi Zhang
BUPT
Beijing, China
zhangx@bupt.edu.cn

Sihong Xie
Lehigh University
Pennsylvania, USA
xiesihong1@gmail.com

ABSTRACT

User-generated product reviews are essential for online platforms like Amazon and Yelp. However, the presence of fake reviews misleads customers. GNN is the state-of-the-art method that detects suspicious reviewers by exploiting the topologies of the graph connecting reviewers, reviews, and products. Nevertheless, the discrepancy in the detection accuracy over different groups of reviewers degrades reviewer engagement and customer trust in the review websites. Unlike the previous belief that the difference between the groups causes unfairness, we study the subgroup structures within the groups that can also cause discrepancies in treating different groups. This paper addresses the challenges of defining, approximating, and utilizing a new subgroup structure for fair spam detection. We first identify subgroup structures in the review graph that lead to discrepant accuracy in the groups. The complex dependencies over the review graph create difficulties in teasing out subgroups hidden within larger groups. We design a model that can be trained to jointly infer the hidden subgroup memberships and exploits the membership for calibrating the detection accuracy across groups. Comprehensive comparisons against baselines on three large Yelp review datasets demonstrate that the subgroup membership can be identified and exploited for group fairness.

KEYWORDS

fairness, spam detection, graphs

ACM Reference Format:
Jiaxin Liu, Yuefei Lyu, Xi Zhang, and Sihong Xie. 2018. Are Your Reviewers Being Treated Equally? Discovering Subgroup Structures to Improve Fairness in Spam Detection. In *Proceedings of 2nd ACM SIGKDD Workshop on Ethical Artificial Intelligence: Methods and Applications (EAI-KDD' 23).* ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Existing fake reviews undermine trust in online commerce websites hosting the reviews. Among all detection methods, graph-based approaches have shown great promise. However, prior methods [9, 10] often prioritize the accuracy or robustness of fraud detectors while overlooking detection fairness. Although some existing works have examined fairness issues in graph-based classifiers concerning sensitive attributes (SAs) like gender, age, and race [1, 7, 16, 18, 19], the anonymity of the spammers hinders the study of fairness problems regarding these traditional SAs. Graph-based spam detectors also suffer from another fairness problem where reviewers are unfairly detected based on their historical post count (degree of the reviewer nodes). Established reviewers and spammers receive lenient scrutiny as their few spam posts remain hidden behind among abundant normal content. New reviewers with minimal posts face a higher risk of false detection and stringent regulation. This discrimination harms user trust and diminishes engagement in online commerce. Formally, this fairness issue stems from variant graph *topologies* [4]. Growing efforts have been made to address fairness concerning topological bias [1, 7, 16, 21], yet few works specifically focus on fairness in the context of graph-based spam detection.

Figure 1 illustrates a review graph comprising user, review, and product nodes. Edges represent the instances where a user post reviews on certain products. Users who post fewer reviews than a specific threshold are labeled as "protected" users, denoted by the SA $A = 1$, while other users are considered "favored" ($A = 0$). When comparing the computational graphs of spams R_1 and R_2, the presence of numerous non-spams dilutes the suspiciousness of R_2 during the bottom-up message passing through GNN's aggregation operation in Eq. (1). Existing users with many reviews can reduce suspiciousness and evade detection, which is unfair to new users.

Maintaining fairness based solely on node degree groups is imprecise because detection fairness also depends on users' ability to hide spam among their normal reviews. Heterogeneous behaviors among favored users, characterized by varying proportions of spam to non-spam reviews, can result in different treatments by the detector. In contrast, protected users, who post a few reviews from the same class, exhibit homogeneous behavior and are treated equally by the detector. Therefore, an additional SA A' is required to accurately describe the heterogeneous behavior of favored users and indicate if their reviews are from the same class. In Figure 1, U_1 (the "mixed" user, denoted by $A' = 1$) posts both spams and non-spams, while U_2 (the "pure" user, denoted by $A' = 0$), posts only spams. U_1 deceives the detector by aggregating messages from non-spams resulting in lower suspiciousness, whereas U_2 does not receive such messages. However, to improve accuracy on the favored group, the GNN unfairly targets spam reviews from pure users like U_2 because they are easier to detect. By enabling the detection of spams from mixed users, the GNN can improve its performance in detecting such spams without negatively impacting the detection accuracy of spams from pure users. Hence, distinguishing mixed users from pure users is crucial. Solving this task involves three challenges:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
EAI-KDD' 23, August 7, 2023, Long Beach, CA, USA
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

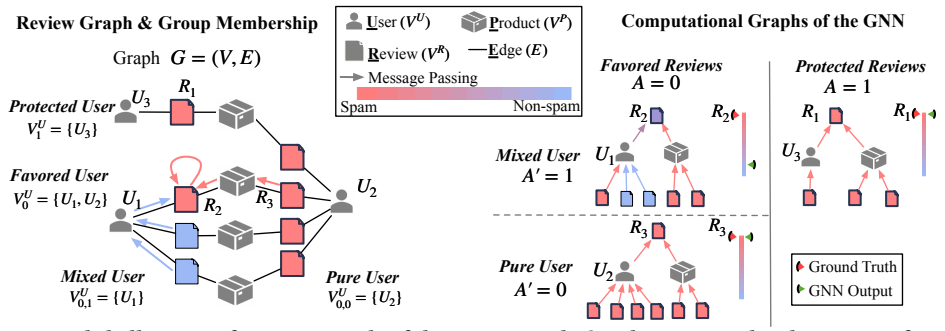


Figure 1: Problem setting and challenges. Left: a toy example of the review graph $\mathcal{G}$ and group membership. GNN infers review suspiciousness by passing and aggregating messages from neighbors. Right: computational graphs of GNN on spam reviews from different (sub-) groups. GNN unfairly assigns a low suspiciousness to spam R_2 posted by mixed user U_1 due to aggregation of messages from non-spams posted by U_1.

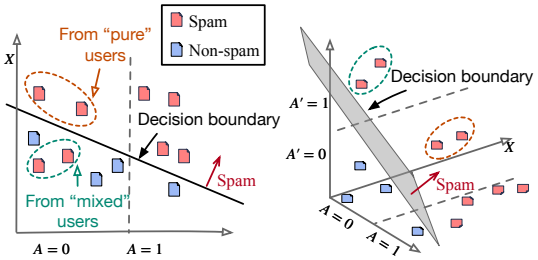


Figure 2: Adding A' improves the accuracy of the detector on spams from the favored group ($A = 0$).

Define the subgroup structure. Previous research primarily focused on well-defined SAs. Some studies, such as [14, 24] divided datasets into groups based on observable SAs and their combinations. Others [2, 6] tackled problems involving unobserved or noisy SAs. These methods were designed for I.I.D vector data with well-defined SAs. In contrast, our approach focuses on enhancing fairness by discovering a novel structural-based SA A' and its approximation. Furthermore, previous works [11, 13] treated SAs as characteristics of the data, not determined by ground truth labels, whereas we aim to identify an undefined SA A' specific to graph-based spam detection and related to the ground truth of reviews.

Infer unknown subgroup membership. We hypothesize that inferring A' helps resolve unfairness among groups by capturing users' heterogeneous behavior. This subgroup indicator A' distinguishes between users in the favored group who genuinely benefit from their non-spam reviews. By incorporating the inferred indicator A', the GNN can achieve a more equitable detection of spams posted by users in the favored group, promoting group fairness (see Figure 2). However, inferring A' is challenging as it relies on unobservable ground truth labels, which are especially difficult to obtain for test data and even for training data due to labeling limitations and the accurate identification of spams being expensive and time-consuming [22, 23].

Promote group fairness through subgroup information. Previous works on fairness consider multiple SAs and formulate optimization problems with fairness constraints for each combination of groups [14, 24]. These SAs must be precise and discrete to categorize the data and ensure fair treatment of discriminated groups. Yet, these premises do not apply to our inferred subgroup membership indicator A', which is probabilistic rather than deterministic. Also, constraints based on thresholding noisy SAs can negatively impact optimization algorithms. Given the sensitivity of fairness optimization to group separation, we avoid using uncertain thresholds to convert the probabilistic membership into a specific group.

Our main contribution is the discovery of a new structural-based and label-related SA A' in fair spam detection on graphs. We define A' and develop a GNN to infer its probability distribution for test users with unlabelled reviews in §3.1. To address the issue of limited training examples, we propose two fairness-aware data augmentation methods to synthesize nodes and edges in §3.2. Additionally, we design a joint training method in §3.3, where the detector utilizes the inferred A'. Experimental results show the presence of unfairness within the favored group, which can be mitigated by leveraging A'. Our method enhances group fairness by accurately inferring the distribution of A' during model training, regardless of the threshold used to split the group based on A.

2 PRELIMINARIES

2.1 Spam detection based on GNN

Our spam detection is based on a review graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of nodes and undirected edges, respectively. Each node $v_i \in \mathcal{V}$ has a feature vector $\mathbf{x}_i$ with node index i. $\mathcal{G}$ contains three types of nodes: user, review, and product. Each node is from only one of the three types. Let $\mathcal{V}^U, \mathcal{V}^R$, and $\mathcal{V}^P$ denote the sets of user, review, and product nodes, respectively. v_i has a set of neighboring nodes denoted by $\mathcal{N}(i) = \{v_j \in \mathcal{V} | e_{i,j} \in \mathcal{E}\}$. The work focuses on detecting spam users and reviews, and the task is essentially a node classification problem. GNN [15] is the state-of-the-art method for node classification. The GNN detector $f_{\boldsymbol{W}}(\cdot)$ learns representation $\mathbf{h}_i^{(l)}$ for node v_i at layer l, where $l = 1, \ldots L$:

$$\mathbf{h}_i^{(l)} = \text{UPDATE}\left(\text{AGGREGATE}\left(\left\{\mathbf{h}_i^{(l-1)}\right\} \bigcup \left\{\mathbf{h}_j^{(l-1)} \mid j \in \mathcal{N}(i)\right\}\right), \boldsymbol{W}^{(l)}\right), \quad (1)$$

where AGGREGATE takes the mean over $\mathbf{h}_i^{(l)}$ and messages passed from its neighboring nodes. UPDATE applies an affine mapping with parameters $\boldsymbol{W}^{(l)}$ followed by a non-linearity (ReLU in $l = 1, \ldots, L-1$ and Sigmoid in $l = L$). The input vector $\mathbf{x}_i = \mathbf{h}_i^{(0)}$ is the representation at layer 0. $\hat{y}_i = \mathbf{h}_i^{(L)} \in \mathbb{R}$ denotes the prediction probability of v_i being spam. The cross-entropy loss is minimized for the training node set $\mathcal{V}^{\text{Tr}}$:

$$\mathcal{L}(\boldsymbol{W}; \mathcal{G}) = \frac{1}{|\mathcal{V}^{\text{Tr}}|} \sum_{v_i \in \mathcal{V}^{\text{Tr}}} \left(-y_i \cdot \log \hat{y}_i - (1 - y_i) \cdot \log(1 - \hat{y}_i)\right), \quad (2)$$

where $y_i \in \{0, 1\}$ is the class for v_i. $\boldsymbol{W}$ represents a collection of parameters in all the layers L. Notations can be found in Appendix B.

2.2 Fairness regularizer

Group. Users $\mathcal{V}^U$ are categorized into the protected group $\mathcal{V}_1^U$, consisting of users with degrees lower than the p-th percentile, and the favored group $\mathcal{V}_0^U$ for the remaining users (see Figure 1 left). The subscript denotes the value of A. Reviews $\mathcal{V}^R$ are assigned to either $\mathcal{V}_1^R$ or $\mathcal{V}_0^R$ based on the group of their associated users.

Fairness regularizer. NDCG is a suitable metric for evaluating the accuracy of the detector due to the highly skewed class distribution. A higher NDCG indicates a more accurate detector. It can also separately measure $\mathcal{V}_0^R$ and $\mathcal{V}_1^R$, allowing the evaluation of group fairness through the NDCG gap (denoted by Δ_{NDCG}). To reduce Δ_{NDCG} while promoting the NDCG of $\mathcal{V}_0^R$ without compromising that of $\mathcal{V}_1^R$, the detector incorporates a fairness regularizer $\mathcal{R}_{\text{fair}}$ which takes negative NDCG of $\mathcal{V}_0^R$. We adopt a differentiable surrogate NDCG [4] for $\mathcal{R}_{\text{fair}}$

$$\mathcal{R}_{\text{fair}}(\boldsymbol{W}; \mathcal{G}) = -\frac{1}{Z_0} \sum_{\substack{i,j: y_i < y_j \\ v_i, v_j \in \mathcal{V}_0^R \cap \mathcal{V}^{\text{Tr}}}} \log\left(1 + \exp\left(\mathbf{h}_j^{(L)} - \mathbf{h}_i^{(L)}\right)\right), \quad (3)$$

$$\mathcal{L}_{\text{GNN}}(\boldsymbol{W}; \mathcal{G}) = \mathcal{L}(\boldsymbol{W}; \mathcal{G}) + \lambda \cdot \mathcal{R}_{\text{fair}}(\boldsymbol{W}; \mathcal{G}), \quad (4)$$

where Z_0 is the total number of pairs of spams and non-spams in the training $\mathcal{V}_0^R$. $\mathcal{L}_{\text{GNN}}$ is the objective function for training $f_{\boldsymbol{W}}(\cdot)$ by adding $\mathcal{R}_{\text{fair}}$ to $\mathcal{L}(\boldsymbol{W}; \mathcal{G})$ in Eq. (2). $\lambda > 0$ is the importance of the fairness regularizer. Note that the fairness regularizer $\mathcal{R}_{\text{fair}}$ regularizes all the models referred to in this paper below.

3 METHODOLOGY

3.1 Subgroup definition and selection

Subgroups. The suspiciousness of spam posted by user v_i decreases as the GNN aggregates non-spams posted by v_i. A new SA A_i' is defined to identify if user v_i benefits from its non-spam reviews:

$$A_i' = \begin{cases} 1 & \text{if } 0 < \sum_{j \in \mathcal{N}(i)} y_j < |\mathcal{N}(i)| \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where $v_i \in \mathcal{V}_0^U$. $\mathcal{N}(i)$ are reviews posted by v_i. Users with $A' = 1$ have spam and non-spam reviews, while users with $A' = 0$ only have reviews belonging to either spams or non-spams. The heterogeneity in behavior leads to the split of $\mathcal{V}_0^U$ into subgroups $\mathcal{V}_{0,0}^U$ and $\mathcal{V}_{0,1}^U$, where subscript denotes the value of A and A' respectively. Since users in $\mathcal{V}_1^U$ primarily post reviews in just one class, A' is unnecessary when $A = 1$.

Infer unobservable subgroup membership. We aim to utilize the new A' to improve the fairness and calibration of the detector across subgroups. Since most review labels are unknown, we employ a GNN $g_{\boldsymbol{\theta}}(\cdot)$ to infer A' for users whose reviews are not fully labeled. In principle, many predictive models can map v_i to A_i'. Still, GNN is chosen for its ability to capture the heterogeneous characteristics of users by modeling the distribution of neighborhood data. Let $\hat{A}_i'$ (A_i', resp.) represent the prediction (ground truth, resp.) of A' for user v_i. The loss of predicting A_i' using $g_{\boldsymbol{\theta}}$ becomes:

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{G}) = \frac{1}{Z_1} \sum_{v_i \in \mathcal{V}^{\text{Tr}} \cap \mathcal{V}^U} (-A_i' \cdot \log \hat{A}_i' - (1 - A_i') \cdot \log(1 - \hat{A}_i')), \quad (6)$$

where Z_1 is the total number of training user nodes.

3.2 Fair Data augmentation

3.2.1 Augmentation for user subgroup $\mathcal{V}_{0,1}^U$. To address the underfitting of $g_{\boldsymbol{\theta}}$ caused by limited training mixed users in our dataset (see Table 3 in Appendix D.1), we synthesize nodes to augment mixed users and their reviews while preserving the original node distribution in $\mathcal{V}_{0,1}^U$. By **replicating** the mixed training users along with their reviews for k times, and slightly perturbing the copies through **pruning** review connections, we ensure diverse but similar representations of the synthetic mixed users compared to the original users. Specifically, oversampling adds multiple copies of the minority without altering the node distribution, while pruning edges connected to non-spam reviews and retaining edges connected to scarce spams helps balance the class of reviews from mixed users (see Figure 3 left).

3.2.2 Augmentation for minority review group $\mathcal{V}_0^R$. It is challenging to train the detector $f_{\boldsymbol{W}}$ with a limited number of favored reviews (see $\%\mathcal{V}_0^R$ in Table 3). We augment the minority group $\mathcal{V}_0^R$ following a graph mixup method [25] that the mixup for GNN inputs, node embeddings at each layer, and labels for the synthetic data become:

$$\tilde{\mathbf{x}}_{ij} = \alpha \mathbf{x}_i + (1 - \alpha)\mathbf{x}_j, \quad \tilde{\mathbf{h}}_{ij}^{(l)} = \alpha \tilde{\mathbf{h}}_i^{(l)} + (1 - \alpha)\tilde{\mathbf{h}}_j^{(l)}, \quad \tilde{y}_{ij} = \alpha y_i + (1 - \alpha) y_j,$$

where $\alpha \in [0, 1]$ and $\tilde{\mathbf{x}}_{ij}$ is the mixture of node attributes $\mathbf{x}_i$ and $\mathbf{x}_j$ at the input layer. $\tilde{\mathbf{h}}_{ij}^{(l)}$ is the mixture at the l-th layer synthesized from the two hidden representations $\tilde{\mathbf{h}}_i^{(l)}$ and $\tilde{\mathbf{h}}_j^{(l)}$ ($\tilde{\mathbf{h}}_{ij}^{(0)} = \tilde{\mathbf{x}}_{ij}$). $\tilde{y}_{ij}$ is the label for $\tilde{\mathbf{x}}_{ij}$.

Our method selectively chooses nodes for mixup to address the class imbalance between majority and minority groups or subgroups in our review graph. We sample the first node from spam reviews in the favored group $\mathcal{V}_0^R$ to ensure that the synthetic reviews resemble the original favored spams and effectively tackle the class imbalance issue in this group. The sampling of the second nodes involves three sets: our method (S_1^{Tr}) and two variants sets (S_0^{Te} and S_1^{Te}) as shown in Figure 3 (right). S_1^{Tr} comprises spams from the protected training group, while S_0^{Te} and S_1^{Te} consist of test reviews from the favored group and the protected group, respectively. The sets for sampling the first and second nodes are:

$$\textbf{Sample first node from: } S_0^{\text{Tr}} = \left\{v_i \mid v_i \in \mathcal{V}_0^R \cap \mathcal{V}^{\text{Tr}}, y_i = 1\right\}. \quad (7)$$

$$\textbf{Sample second node from one of: } S_1^{\text{Tr}} = \left\{v_j \mid v_j \in \mathcal{V}_1^R \cap \mathcal{V}^{\text{Tr}}, y_j = 1\right\},$$
$$S_0^{\text{Te}} = \left\{v_j \mid v_j \in \mathcal{V}_0^R \cap \mathcal{V}^{\text{Te}}\right\}, \quad S_1^{\text{Te}} = \left\{v_j \mid v_j \in \mathcal{V}_1^R \cap \mathcal{V}^{\text{Te}}\right\}. \quad (8)$$

Since the labels of reviews sampled from S_0^{Te} and S_1^{Te} are unknown, the synthetic review has the same label as the first node, i.e., $\tilde{y}_{ij} = 1$.

3.3 Joint model

3.3.1 Utilizing subgroup membership. Incorporating subgroup membership into the objective function typically involves adding a fairness regularizer that operates on the subgroups defined by the inferred membership. However, existing fairness regularizers assume deterministic group membership rather than the probabilistic estimation provided by $g_{\boldsymbol{\theta}}$. We consider the inferred $\hat{A}'$ as a supplementary attribute that informs the detector about subgroup membership and the uncertainty associated with its estimation. As

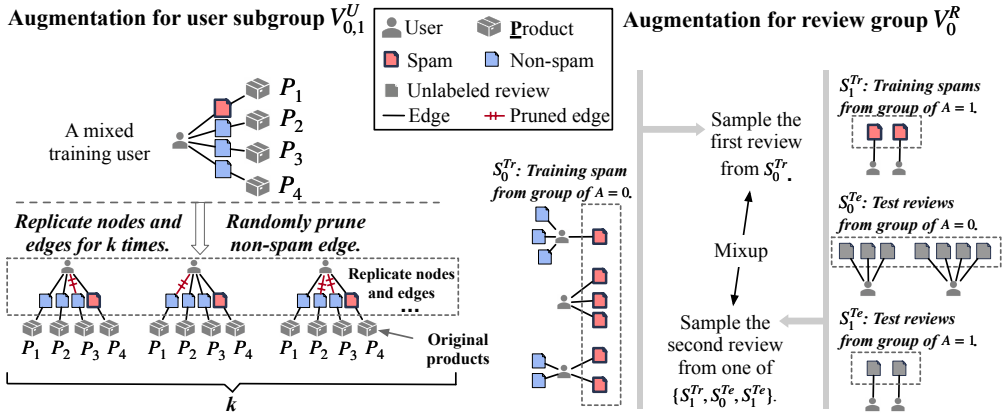


Figure 3: Toy example of our proposed fair augmentation methods.

a result, the user's feature vector is expanded to include $\hat{A}'$, resulting in $\mathbf{x}' = [\mathbf{x}, \hat{A}'] \in \mathbb{R}^{d+1}$, where $\hat{A}'$ represents the probability of the user having $A' = 1$ as predicted by $g_{\boldsymbol{\theta}}$.

3.3.2 Optimization for the joint model. We propose a joint optimization approach for two GNNs $g_{\boldsymbol{\theta}}$ and $f_{\boldsymbol{W}}$, allowing them to *co-adopt* to each other. The inferred $\hat{A}'$ is treated as a function of $\boldsymbol{\theta}$, denoted as $\hat{A}'(\boldsymbol{\theta})$, rather than a constant. Consequently, the expanded user feature $\mathbf{x}'(\boldsymbol{\theta}) = [\mathbf{x}, \hat{A}'(\boldsymbol{\theta})]$ incorporates $\boldsymbol{\theta}$ as parameters. The loss for optimizing the detector $f_{\boldsymbol{W}}$ in Eq. (4) becomes $\mathcal{L}_{\text{GNN}}(\boldsymbol{W}, \boldsymbol{\theta}; \mathcal{G})$. By connecting the computational graph between $\boldsymbol{W}$ and $\boldsymbol{\theta}$, $\boldsymbol{\theta}$ receives gradients from $f_{\boldsymbol{W}}$ to improve the accuracy of inferring A'. Simultaneously, the updated $\boldsymbol{\theta}$ is used to enhance the performance of $f_{\boldsymbol{W}}$. During training, $\boldsymbol{W}$ and $\boldsymbol{\theta}$ are updated jointly:

$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \beta_1 \nabla_{\boldsymbol{W}} \mathcal{L}_{\text{GNN}}(\boldsymbol{W}, \boldsymbol{\theta}; \mathcal{G}) \tag{9}$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \underbrace{\beta_2 \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}; \mathcal{G})}_{\text{Gradient from } g_{\boldsymbol{\theta}}} - \underbrace{\beta_2 \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{GNN}}(\boldsymbol{W}, \boldsymbol{\theta}; \mathcal{G})}_{\text{Gradient from } f_{\boldsymbol{W}}} \tag{10}$$

where β_1 (β_2) is the learning rate for updating $\boldsymbol{W}$ ($\boldsymbol{\theta}$). Please refer to Algorithm 1 in Appendix C for a complete description.

4 EXPERIMENTS

Our research aims to address the following research questions: **RQ1**: Are there fairness issues between the favored and protected groups and between the mixed and pure groups when using a GNN spam detector? Can the inferred A' improve group fairness? **RQ2**: How can we infer the subgroup membership A' when there is a shortage of training examples? **RQ3**: Can the joint training method effectively enhance both the AUC of predicted A' and group fairness simultaneously? **RQ4**: Does the accuracy of predicting A' contribute to the improvement in group fairness?

4.1 Experimental Settings

4.1.1 Datasets. We used three Yelp review datasets (referred to as "Chi", "NYC", and "Zip", see Table 3). For each dataset, user nodes are divided into the favored group (top p% high-degree users, $A = 0$) and the protected group (the remaining users, $A = 1$) by the percentile p in Section 2.2. More statistical information about datasets can be found in Appendix D.1.

4.1.2 Evaluation Metrics. NDCG was used to assess the accuracy of the detector. A higher NDCG indicates that the detector assigns higher suspiciousness values to spams than non-spams. We introduced a metric called "**Ave**rage **F**alse **R**anking **R**atio" (AFRR) to evaluate the ranking of spams within the group of $A = 0$. AFRR measures the average relative ranking between spams from mixed and pure users. A lower AFRR means fewer non-spams ranked higher than spams. AUC was utilized to evaluate the performance of the second GNN $g_{\boldsymbol{\theta}}$ in predicting A'. A higher AUC indicates better predictive performance.

Refer to Appendix D.1 for more details, including the formal definition of AFRR and the characteristics of each metric.

4.1.3 Methods compared. **Joint+GNN-S_1^{Tr}** denotes our method: **Joint** refers to the joint training for two GNNs ($f_{\boldsymbol{W}}$ and $g_{\boldsymbol{\theta}}$ in Section 3.3), **GNN-S_1^{Tr}** refers to a GNN with our mixup method in Eq. (8). "a+b" represents a combination between method "a" for obtaining the value of A' and method "b", a spam detector.

Baselines for obtaining the value of A' (selections for "a"): **w/o** does not involve A'. **Random** randomly assigns 1/0 to A'. **GT** assigns the ground truth of A' for users, which is the ideal case, as A' is unknown. **Pre-trained** is a variant of Joint that pre-trains $g_{\boldsymbol{\theta}}$ to infer A' and fixes the inferred A' when training $f_{\boldsymbol{W}}$.

Baselines for the spam detectors (selections for "b"): **FairGNN** [7] is an adversarial method that aims to achieve fair predictions for all groups defined by the known A. **EDITS** [8] modifies the node attribute and the graph structure to debias the graph. FairGNN and EDITS consider A as a known SA and exclude any information about A' defined by our work. **GNN** is the vanilla GNN. **GNN-S_0^{Te}** is a GNN with the mixup Case 2 in Eq. (8). **GNN-S_1^{Te}** is a GNN with the mixup Case 3 in Eq. (8).

4.2 Results

Due to page limitations, we only include the results for answering **RQ1**, with a focus on the percentile $p = 20$. Additional results for answering **RQs 2-4** and using percentiles $p = \{15, 10\}$ can be found in Appendix D.2.

Group Fairness. To answer **RQ1**, we take the gap in NDCGs values between groups of $A = 0$ and $A = 1$ as the fairness metric, denoted by Δ_{NDCG}. Table 1 presents the NDCG values for the outputs of

Table 1: NDCG values for spam detectors on three Yelp datasets are shown with a 20-percentile cutoff for defining groups of A based on user degree. The table displays the mean and standard deviation of NDCG scores for all reviews (NDCG($\mathcal{V}^R$) ↑), for the group with $A = 1$ (NDCG($\mathcal{V}_1^R$) ↑), and the NDCG difference between the two groups (Δ_{NDCG} ↓) across all ten splits. "↑" means the larger, the better; "↓" means the opposite. Our method is denoted by "*". A smaller Δ_{NDCG} indicates a fairer model.

Detector	Metrics(%)	Chi	NYC	Zip
FairGNN [7]	NDCG($\mathcal{V}^R$) ↑	86.2±0.1	85.9±0.0	89.6±0.5
	NDCG($\mathcal{V}_1^R$) ↑	86.2±0.1	86.0±0.0	89.6±0.4
	Δ_{NDCG} ↓	53.7±2.1	20.7±1.1	36.4±4.6
EDITS [8]	NDCG($\mathcal{V}^R$) ↑	84.3±0.3	84.9±0.1	89.2±0.0
	NDCG($\mathcal{V}_1^R$) ↑	84.3±0.3	85.1±0.1	89.3±0.0
	Δ_{NDCG} ↓	50.9±2.3	32.7±1.5	39.4±10.6

Detector	Metrics(%)	GNN(g_θ) w/o	GNN(g_θ) Pre-trained	GNN(g_θ) Joint*	GNN(g_θ) w/o	GNN(g_θ) Pre-trained	GNN(g_θ) Joint*	GNN(g_θ) w/o	GNN(g_θ) Pre-trained	GNN(g_θ) Joint*
GNN	NDCG($\mathcal{V}^R$) ↑	84.5±0.9	83.2±1.5	83.3±2.2	85.2±0.8	85.1±0.8	85.2±0.5	88.4±1.4	87.6±1.5	88.6±1.0
	NDCG($\mathcal{V}_1^R$) ↑	84.7±0.9	83.5±1.3	83.6±2.1	85.1± 0.8	85.3±0.8	85.2±0.5	88.4±1.4	87.6±1.5	88.6±1.0
	Δ_{NDCG} ↓	51.2±2.1	51.0±3.0	**50.7**±3.5	21.9±7.0	21.8±7.8	**21.3**±8.9	36.3±10.4	**34.3**±10.8	34.8±11.1
GNN-S$_1^{\text{Tr}}$*	NDCG($\mathcal{V}^R$) ↑	85.6±0.7	85.8±0.5	85.6±0.8	85.8±0.1	85.9±0.0	85.9±0.0	89.7±0.2	89.7±0.1	89.6±0.1
	NDCG($\mathcal{V}_1^R$) ↑	85.6±0.7	85.8±0.5	85.7±0.8	85.9±0.0	86.0±0.0	86.0±0.0	89.7±0.2	89.7±0.1	89.6±0.1
	Δ_{NDCG} ↓	51.6±0.9	50.3±1.0	**50.1**±1.0	19.1±5.5	19.0±5.2	**17.9**±6.1	38.7±7.2	36.0±9.0	**34.3**±11.5
GNN-S$_0^{\text{Te}}$	NDCG($\mathcal{V}^R$) ↑	85.1±0.7	85.3±1.4	85.2±1.4	85.3±0.6	85.4±0.5	85.4±0.4	89.4±0.6	89.6±0.1	89.0±0.06
	NDCG($\mathcal{V}_1^R$) ↑	85.2±0.8	83.4±1.3	83.5±1.3	85.2±0.6	85.5±0.4	85.5±0.4	89.4±0.06	89.0±0.01	89.1±0.06
	Δ_{NDCG} ↓	51.2±1.5	**50.9**±2.4	**50.9**±2.3	21.9±6.9	21.9±6.3	**20.9**±9.4	38.9±7.6	36.9±9.5	**34.9**±11.0
GNN-S$_1^{\text{Te}}$	NDCG($\mathcal{V}^R$) ↑	84.7±1.3	83.7±0.9	83.1±0.9	85.7±0.1	85.8±0.1	85.8±0.2	89.6±0.3	89.6±0.1	89.5±0.3
	NDCG($\mathcal{V}_1^R$) ↑	84.8±1.3	83.9±0.9	83.4±0.9	85.8±0.1	85.8±0.1	85.8±0.1	89.6±0.3	89.6±0.1	89.5±0.3
	Δ_{NDCG} ↓	51.3±0.6	50.8±0.6	**50.2**±1.0	21.0±5.4	19.8±5.4	**19.3**±6.8	38.7±7.2	36.2±9.5	**34.6**±11.4

various spam detectors using 20th percentile of user node degrees as the cutoff for groups of A. The table includes detectors grouped into two sections. The upper section consists of **FairGNN** and **EDITS**, which do not consider A' defined in our work. The lower section includes four detectors that consider A' within each dataset column representing three methods for obtaining the value of A'.

FairGNN, **EDITS**, and **w/o+GNN** detectors exhibit significant Δ_{NDCG} values, indicating the presence of a widespread fairness issue in the spam-detection task on the graph by applying GNN-based fair models. Notably, **FairGNN** and **EDITS** have larger Δ_{NDCG} values, implying that their improvements in NDCGs favor the favored group more than the protected group, exacerbating fairness concerns. In contrast, for detectors in the lower section, the proposed **Joint** method consistently demonstrates the smallest Δ_{NDCG} in most cases.

It is worth noting that in **Joint**, g_θ receives an additional gradient from f_W, as indicated by Eq. (10). However, for detectors without our fair data augmentation (i.e., **Joint+GNN**), this additional gradient may cause g_θ to infer A' and negatively impact the performance of f_W. Among the methods for obtaining the value of A', the detector with our augmentation **GNN-S$_1^{\text{Tr}}$** consistently shows the smallest Δ_{NDCG} in almost all cases. This suggests that maintaining the original distribution while performing the mixup method is more challenging in datasets with fewer mixed users, such as Chi and Zip.

5 CONCLUSION

This work addresses fairness in a graph-based spam detection task, specifically focusing on the unfairness between the protected and favored groups defined by the known SA node degree. To capture the heterogeneous behaviors of the favored users, A' is introduced, dividing favored users into mixed and pure categories. The value of A' for test users is inferred using a second GNN g_θ and integrated as a supplementary feature feed into the detector f_W. Our proposed **Joint** method simultaneously improves detector fairness and enhances the quality of inferred A'. The experimental results on three Yelp datasets, incorporating fair data augmentation, validate the effectiveness of the **Joint** method. Our approach successfully promotes group fairness by enabling the detector to enhance the suspiciousness of spam from both pure and mixed users.

ACKNOWLEDGMENTS

Sihong Xie was supported in part by the National Science Foundation under NSF Grants IIS-1909879, CNS-1931042, IIS-2008155, and IIS-2145922."

REFERENCES

[1] Agarwal, C., Lakkaraju, H., Zitnik, M.: Towards a unified framework for fair and stable graph representation learning. In: UAI (2021)
[2] Awasthi, P., Kleindessner, M., Morgenstern, J.: Equalized odds postprocessing under imperfect group information. In: AISTATS (2020)
[3] Bose, A., Hamilton, W.: Compositional fairness constraints for graph embeddings. In: ICML (2019)
[4] Burkholder, K., Kwock, K., Xu, Y., Liu, J., Chen, C., Xie, S.: Certification and trade-off of multiple fairness criteria in graph-based spam detection. In: CIKM (2021)
[5] Buyl, M., De Bie, T.: Debayes: a bayesian method for debiasing network embeddings. In: ICML. PMLR (2020)
[6] Celis, L.E., Huang, L., Keswani, V., Vishnoi, N.K.: Fair classification with noisy protected attributes: A framework with provable guarantees. In: ICML (2021)
[7] Dai, E., Wang, S.: Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information. In: WSDM (2021)
[8] Dong, Y., Liu, N., Jalaian, B., Li, J.: Edits: Modeling and mitigating data bias for graph neural networks. In: WWW (2022)

[9] Dou, Y., Liu, Z., Sun, L., Deng, Y., Peng, H., Yu, P.S.: Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In: CIKM (2020)
[10] Dou, Y., Ma, G., Yu, P.S., Xie, S.: Robust spammer detection by nash reinforcement learning. In: SIGKDD (2020)
[11] Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.: Fairness through awareness. In: ITCS (2012)
[12] Feng, W., Zhang, J., Dong, Y., Han, Y., Luan, H., Xu, Q., Yang, Q., Kharlamov, E., Tang, J.: Graph random neural networks for semi-supervised learning on graphs. NeurIPS (2020)
[13] Hardt, M., Price, E., Srebro, N.: Equality of opportunity in supervised learning. NeurIPS (2016)
[14] Kearns, M., Neel, S., Roth, A., Wu, Z.S.: Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In: ICML (2018)
[15] Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. ICLR (2017)
[16] Li, P., Wang, Y., Zhao, H., Hong, P., Liu, H.: On dyadic fairness: Exploring and mitigating bias in graph connections. In: ICLR (2021)
[17] Loveland, D., Pan, J., Bhathena, A.F., Lu, Y.: Fairedit: Preserving fairness in graph neural networks through greedy graph editing. arXiv preprint arXiv:2201.03681 (2022)
[18] Ma, J., Deng, J., Mei, Q.: Subgroup generalization and fairness of graph neural networks. NeurIPS (2021)
[19] Rahman, T., Surma, B., Backes, M., Zhang, Y.: Fairwalk: Towards Fair Graph Embedding. In: IJCAI-19 (2019). https://doi.org/10.24963/ijcai.2019/456
[20] Rong, Y., Huang, W., Xu, T., Huang, J.: Dropedge: Towards deep graph convolutional networks on node classification. arXiv preprint arXiv:1907.10903 (2019)
[21] Spinelli, I., Scardapane, S., Hussain, A., Uncini, A.: Fairdrop: Biased edge dropout for enhancing fairness in graph representation learning. TAI (2021)
[22] Tan, E., Guo, L., Chen, S., Zhang, X., Zhao, Y.: Unik: Unsupervised social network spam detection. In: CIKM (2013)
[23] Tian, Y., Mirzabagheri, M., Tirandazi, P., Bamakan, S.M.H.: A non-convex semi-supervised approach to opinion spam detection by ramp-one class svm. Information Processing & Management (2020)
[24] Ustun, B., Liu, Y., Parkes, D.: Fairness without harm: Decoupled classifiers with preference guarantees. In: ICML (2019)
[25] Wang, Y., Wang, W., Liang, Y., Cai, Y., Hooi, B.: Mixup for node and graph classification. In: WWW (2021)
[26] You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., Shen, Y.: Graph contrastive learning with augmentations. NeurIPS (2020)
[27] Zhao, T., Liu, G., Günnemann, S., Jiang, M.: Graph data augmentation for graph machine learning: A survey. arXiv preprint arXiv:2202.08871 (2022)
[28] Zhao, T., Liu, Y., Neves, L., Woodford, O., Jiang, M., Shah, N.: Data augmentation for graph neural networks. In: AAAI (2021)

A RELATED WORK

A.1 Fairness on graphs.

Fairness on graphs has been explored from various perspectives. Researchers have aimed to achieve fairness in node embeddings, representations, and classification results regarding discrete and well-defined sensitive attributes [5, 19]. Adversarial frameworks have been introduced to mitigate unfairness biases related to sensitive attribute [1, 3, 7], where fairness regularizers are incorporated to ensure insensitivity of the model towards those attributes. Methods like FairGNN [7] employ adversarial debiasing by leveraging an additional GNN to estimate missing sensitive information. Other approaches modify graph topology or edge weights to obtain fair node embeddings or predictions. FairAdj [16] adjusts edge connections and learns a fair adjacency matrix by adding contains for graph structure. FairEdit [17] proposes model-agnostic algorithms which perform edge addition and deletion using the gradient of fairness loss. FairDrop [21] addresses bias by excluding biased edges to counteract homophily. However, these methods typically rely on known or well-defined sensitive attributes. Some studies have also investigated fairness problems involving multiple sensitive attributes, which may lead to fairness violations when enforcing fairness simultaneously across all attributes. Approaches like those in [14, 24] have designed classifiers with multiple fairness constraints generated from combinations of sensitive attributes.

A.2 Augmentation on the graph.

Graph augmentation has gained significant attention in recent years [27]. Studies such as [12, 25] have focused on augmenting graphs at the node level, generating synthetic data through techniques like node mixup or node removal. Additionally, graph augmentation has been performed at the edge level, including modifications such as adding orr removing edges, either deterministically [28] or stochastically [20]. Moreover, some methods augment graph data at the node feature level by randomly masking node features [26].

B NOTATION

Table 2 provides a summary of the key notations used throughout this paper.

Table 2: Notations and definitions.

Notations	Definitions
Graph notations	
$\mathcal{G}$	Review graph
$\mathcal{V}, \mathcal{E}$	Nodes and Edges of graph $\mathcal{G}$
$\mathbf{x}_i, y_i$	Feature and label of node v_i
$\mathcal{N}(i)$	Set of direct neighbors of v_i
$\lvert\mathcal{V}\rvert$	Cardinality of a set $\mathcal{V}$
$\mathcal{V}^{\text{Tr}}, \mathcal{V}^{\text{Te}}$	Training nodes and test nodes
$\mathcal{V}^U, \mathcal{V}^R, \mathcal{V}^P$	User, review, and product nodes
Group notations	
A, A'	Binary sensitive attributes (0/1)
$\mathcal{V}^R_a, \mathcal{V}^U_a$	Review and user nodes from group of $A = a$
$\mathcal{V}^R_{a,a'}, \mathcal{V}^U_{a,a'}$	Review and user nodes from group of $A = a$ and $A' = a'$
Model notations	
$f_{\boldsymbol{W}}(\cdot), g_{\boldsymbol{\theta}}(\cdot)$	GNNs with parameters $\boldsymbol{W}$ and $\boldsymbol{\theta}$
$\hat{y}_i, \hat{A}'_i$	Output of $f_{\boldsymbol{W}}(\cdot), g_{\boldsymbol{\theta}}(\cdot)$ for v_i
$\mathbf{h}_i^{(l)}$	Representation of v_i on layer l
$\tilde{\mathbf{x}}_{ij}$	Synthetic node by mixing-up v_i and v_j
$\tilde{y}_{ij}$	Label for the synthetic node $\tilde{\mathbf{x}}_{ij}$

Algorithm 1 Joint training for $g_{\boldsymbol{\theta}}$ and $f_{\boldsymbol{W}}$.

Input: graph $\mathcal{G}$; node features $\boldsymbol{X}$; sensitive attribute A; training epochs T; hyper-parameter λ, learning rates β_1 and β_2, and replication times k.
Output: optimal $\boldsymbol{\theta}$ and $\boldsymbol{W}$ of $g_{\boldsymbol{\theta}}$ and $f_{\boldsymbol{W}}$, respectively.
Initialize parameters $\boldsymbol{\theta}$ and $\boldsymbol{W}$.
Replicate users and reviews for k times as in Section 3.2.1. ▷ Augmentation for $\mathcal{V}^U_{0,1}$.
for $t = 1, \ldots, T$ **do**
  Prune non-spam edges as in Section 3.2.1. ▷ Add data variations
  Infer $\Pr(A' = 1)$ for users using $g_{\boldsymbol{\theta}}$.
  Concatenate $\hat{A}'$ to user feature vectors as in Section 3.3.
  Mixup between two reviews sampled from S_0^{Tr} as in Eq. (7) and one of $\{S_1^{\text{Tr}}, S_0^{\text{Te}}, S_1^{\text{Te}}\}$ as in Eq. (8). ▷ Augmentation for $\mathcal{V}^R_0$.
  Update $\boldsymbol{W}$ and $\boldsymbol{\theta}$ following Eq. (9) and (10).
end for

C ALGORITHM

D ADDITIONAL RESULTS

D.1 Experimental Details

D.1.1 Dataset and Setups. The Yelp datasets ("Chi", "NYC", and "Zip') are commonly used in previous spam detection tasks [4, 9], containing three types of node: products, reviews, and users (see Table 3). To study fairness spam detection regarding the degree of user nodes, we set a cutoff degree of user nodes (p-th percentile in Section 2.2). We conduct experiments with $p \in \{20, 15, 10\}$ treated as a hyper-parameter distinguishing favored (top p% high-degree users, $A = 0$) from protected groups (the remaining users, $A = 1$). Reviews have the same value of A as their associated users. The favored users are further split into pure ($A' = 0$) and mixed ($A' = 1$) users following Eq. (5). Users are divided into training (30%), validation (10%), and test (60%) sets with their associated reviews.

D.1.2 Evaluation Metrics. For evaluating the group fairness, we begin by calculating the NDCG score on group of $\mathcal{V}_1^R$ and $\mathcal{V}_0^R$, denoted as NDCG($\mathcal{V}_1^R$) and NDCG($\mathcal{V}_0^R$), respectively. Then, the group fairness can be measured by the NDCG gap

$$\Delta_{\text{NDCG}} = \text{NDCG}(\mathcal{V}_1^R) - \text{NDCG}(\mathcal{V}_0^R). \tag{11}$$

Note that the GNN detector always achieves better performance on $\mathcal{V}_1^R$, i.e., $\Delta_{\text{NDCG}} > 0$ holds all the time. A smaller Δ_{NDCG} indicates fairer detection results for two groups involved.

"**A**verag**e** **F**alse **R**anking **R**atio" is designed to effectively evaluate the intricate ranking performance inside the group of $A = 0$. AFRR computes the average of relative ranking between spams from

Table 3: Statistics of the datasets. We list the numbers of products, reviews, and users with the proportion of favored users/reviews ($\%\mathcal{V}_0^U/\%\mathcal{V}_0^R$) and mixed users/reviews ($\%\mathcal{V}_{0,1}^U/\%\mathcal{V}_{0,1}^R$) under 20-th, 15-th, and 10-th percentile (PC) of cutoff degree of the user groups. The last column gives the ratio of spam in the group of $A = 0$ to $A = 1$.

Name	Data Statistics						$\frac{P(Y=1\|A=0)}{P(Y=1\|A=1)}$
	$\|\mathcal{V}^P\|$	$\|\mathcal{V}^R\|$		$\|\mathcal{V}^U\|$			
Chi	201	67,395		38,063			
		PC	$\%\mathcal{V}_0^R$	$\%\mathcal{V}_{0,1}^R$	$\%\mathcal{V}_0^U$	$\%\mathcal{V}_{0,1}^U$	
		20th	13.581%	0.116%	1.781%	0.011%	0.0438
		15th	18.687%	0.129%	3.003%	0.026%	0.0125
		10th	27.003%	0.224%	5.735%	0.058%	0.0282
NYC	923	358,911		160,220			
		PC	$\%\mathcal{V}_0^R$	$\%\mathcal{V}_{0,1}^R$	$\%\mathcal{V}_0^U$	$\%\mathcal{V}_{0,1}^U$	
		20th	12.665%	0.193%	0.760%	0.009%	0.0479
		15th	16.264%	0.258%	1.171%	0.018%	0.0289
		10th	23.096%	0.360%	2.260%	0.034%	0.0420
Zip	5,044	608,598		260,277			
		PC	$\%\mathcal{V}_0^R$	$\%\mathcal{V}_{0,1}^R$	$\%\mathcal{V}_0^U$	$\%\mathcal{V}_{0,1}^U$	
		20th	6.859%	0.050%	0.272%	0.002%	0.0426
		15th	15.658%	0.145%	1.020%	0.009%	0.0139
		10th	22.342%	0.278%	1.968%	0.018%	0.0241

mixed and pure users

$$\text{AFRR}_{A'} = \frac{1}{Z} \sum_{\substack{y_j=1 \\ v_j \in \mathcal{V}_{0,A'}^R}} \frac{\sum_{i=1}^{|\mathcal{V}_0^R|} \mathbb{1}\left[\hat{y}_i > \hat{y}_j, y_i = 0\right]}{\sum_{i=1}^{|\mathcal{V}_0^R|} \mathbb{1}\left[y_i = 0\right]}, \quad A' \in \{0,1\} \tag{12}$$

where $A' \in \{0,1\}$ denotes the subgroup membership. Z is the number of spams from a subgroup. $\mathcal{V}_{0,A'}^R$ denotes the reviews from a subgroup users. The ratio in the above equation calculates the proportion of non-spams ranked higher than spams over all the non-spams from the group of $A = 0$. The lower the AFRR, the fewer non-spams ranked higher than spams. Compared to NDCG, AFRR considers the non-spams across different subgroups and ignores the relative ranking of spams from the other subgroup.

Since there is a second GNN $g_{\boldsymbol{\theta}}$, AUC is employed to evaluate the performance of $g_{\boldsymbol{\theta}}$ in predicting A'. The larger the AUC value, the more accurate A' given by $g_{\boldsymbol{\theta}}$.

D.1.3 Hyperparameter setting. We set $T = 300$, $\lambda = 5$, $\beta_1 = \beta_2 = 0.001$, weight decay = 0.0001 for both $f_{\boldsymbol{W}}$ and $g_{\boldsymbol{\theta}}$ in Algorithm 1, mixup weight $\alpha = 0.8$. There are 10 training-validation-test splits of the three datasets, and all the results are based on the aggregated performance of all splits.

D.2 Results

D.2.1 Group Fairness. Like the big table given in the main paper, we present the NDCG scores when setting the percentile $p = \{15, 10\}$ in Table 4 and 5. Still, **FairGNN** and **EDITS** give relatively large Δ_{NDCG} in two tables, demonstrating the presence of the fairness issue. Also, our method achieves the smallest Δ_{NDCG} among all the methods in most cases.

D.2.2 Explanation of improved group fairness. Rather than simply obtaining a fair spam detector towards favored and protected groups, we also want to verify the effectiveness of introducing A' in mitigating this intra-group fairness issue. Figure 6 presents the test AFRRs of pure and mixed subgroups across four methods {**w/o+GNN**, **Joint+GNN-S$_1^{\text{Tr}}$**, **Joint+GNN-S$_0^{\text{Te}}$**, **Joint+ GNN-S$_1^{\text{Te}}$**}. It demonstrates the impact of adding A' on spams from subgroups and resulting improvements in NDCG for the protected group. **w/o+GNN** reveals that spams from mixed users have larger AFRRs compared to the pure users across all datasets, indicating the basic GNN tends to rank spams from pure users higher than those from mixed users within the favored group. By introducing A' and employing the fair augmentation methods, the AFRR is reduced for mixed users and occasionally for pure users. Our method (rightmost) improves the NDCG for the protected group primarily by elevating the ranking of spams from mixed users and sometimes from pure users.

D.2.3 Evaluation of the Joint method on improving the quality of A'. To answer **RQ2** and **RQ3**, we investigate the relationship between group fairness and the quality of inferred A'. In Table 1, 4, and 5, we observe that **Joint** method generally exhibits smaller Δ_{NDCG} compared to **Pre-trained**. To gain a deeper understanding of the advantages of **Joint**, we examine the AUC gap of A' (x-axis), as estimated by $g_{\boldsymbol{\theta}}$, plotted against the corresponding Δ_{NDCG} difference (y-axis) in Figure 4. Most models in the area I indicate that **Joint** simultaneously promotes the accuracy of $g_{\boldsymbol{\theta}}$ and the fairness of $f_{\boldsymbol{W}}$. Since **Joint** updates $\boldsymbol{\theta}$ using the additional gradient coming from $f_{\boldsymbol{W}}$ (see Eq. (10)), our fair mixup strategies effectively mitigate the overfitting for $g_{\boldsymbol{\theta}}$ with more gradients from the synthetic data.

D.2.4 Impact of the accurate A' on group fairness. Given the correlation between the quality of A' and group fairness, we further investigate this relationship to answer **RQ4** by manipulating the level of noise in A'. To assess this correlation, we introduce methods that either increase (i.e., **Random** method) or decrease (i.e., **GT** method) the noise in A'. Figure 7 presents the corresponding Δ_{NDCG} for detectors employing different approaches to assign values to A', where the x-axis represents the reduction in noise, progressing from left to right. Notably, as the detector obtains a more accurate inference of the values of A', the Δ_{NDCG} decreases.

D.2.5 Sensitivity studies for the replication times k and if pruning non-spam edges. Figure 5 illustrates the test AUCs of $g_{\boldsymbol{\theta}}$ for different replications values $k = \{50, 100\}$ and the effect of pruning non-spams edges (as discussed in Section 3.2.1). Pruning generally yields better AUCs compared to no pruning, except for the case of $k = 100$ on the Zip dataset. Furthermore, the AUCs for the Chi and Zip datasets tend to decrease as the value of k increases. The sensitivity can be attributed to the limited presence of mixed users in Chi and Zip, making it more challenging to effectively mimic the original node distribution with synthesized data, thereby leading to overfitting of $g_{\boldsymbol{\theta}}$. Hence, the choice of k is highly related to the dataset distribution. By evaluating the validation set, we determine that the optimal replication values are $k = 100$ for NYC and $k = 50$ for Chi and Zip.

Table 4: NDCG values for the outputs of detectors on Yelp datasets with a user degree cutoff at the 15-percentile for defining the group of A. The table displays the mean and standard deviation of NDCG scores for all reviews (NDCG($\mathcal{V}^R$) ↑), for the group with $A = 1$ (NDCG($\mathcal{V}_1^R$) ↑), and the NDCG difference between the two groups (Δ_{NDCG} ↓) across all ten splits. "↑" means the larger, the better; "↓" means the opposite. Our method is denoted by "*". A smaller Δ_{NDCG} indicates a fairer model.

Detector	Metrics(%)	Chi	NYC	Zip
FairGNN [7]	NDCG($\mathcal{V}^R$) ↑	86.2±0.2	85.9±0.1	89.9±0.0
	NDCG($\mathcal{V}_1^R$) ↑	86.3±0.2	86.0±0.1	90.0±0.0
	Δ_{NDCG} ↓	45.8±1.7	20.0±2.4	31.0±2.7
EDITS [8]	NDCG($\mathcal{V}^R$) ↑	84.3±0.2	85.0±0.1	89.2±0.0
	NDCG($\mathcal{V}_1^R$) ↑	84.4±0.2	85.1±0.1	89.3±0.0
	Δ_{NDCG} ↓	43.8±2.6	32.4±1.5	34.4±3.5

Detector	Metrics(%)	GNN(g_θ) w/o	GNN(g_θ) Pre-trained	GNN(g_θ) Joint*	GNN(g_θ) w/o	GNN(g_θ) Pre-trained	GNN(g_θ) Joint*	GNN(g_θ) w/o	GNN(g_θ) Pre-trained	GNN(g_θ) Joint*
GNN	NDCG($\mathcal{V}^R$) ↑	84.3±1.3	84.3±0.9	84.4±0.9	85.7±0.2	84.5±0.4	84.6±0.5	89.5±0.2	88.9±0.6	88.9±0.6
	NDCG($\mathcal{V}_1^R$) ↑	84.6±1.1	84.7±0.7	84.8±0.7	85.8± 0.2	84.6±0.4	84.6±0.5	89.5±0.2	88.9±0.6	88.9±0.7
	Δ_{NDCG} ↓	41.8±2.4	**40.9**±3.7	**40.9**±4.0	16.3±3.9	15.8±2.7	**15.2**±6.1	26.1±3.4	26.6±2.5	**25.9**±2.7
GNN-S_1^{Tr}*	NDCG($\mathcal{V}^R$) ↑	86.0±0.4	85.9±0.3	85.9±0.2	85.9±0.1	85.9±0.1	85.9±0.1	89.7±0.1	89.9±0.0	87.9±2.1
	NDCG($\mathcal{V}_1^R$) ↑	86.0±0.4	86.0±0.3	86.0±0.2	85.9±0.1	85.9±0.1	85.9±0.1	89.8±0.1	89.9±0.0	87.9±2.1
	Δ_{NDCG} ↓	43.2±2.0	41.9±5.8	**40.5**±5.1	16.5±3.8	**15.6**±2.9	**15.6**±3.6	26.4±3.3	27.0±3.2	**22.7**±4.2
GNN-S_0^{Te}	NDCG($\mathcal{V}^R$) ↑	84.4±1.5	84.3±1.1	84.4±0.9	85.8±0.2	84.7±0.4	84.7±0.4	89.4±0.2	89.1±0.6	89.1±0.6
	NDCG($\mathcal{V}_1^R$) ↑	84.7±1.1	84.6±0.9	84.8±0.	85.9±0.2	84.7±0.5	84.7±0.5	89.5±0.2	89.1±0.6	89.2±0.6
	Δ_{NDCG} ↓	43.4±2.4	41.7±4.6	**41.1**±4.2	16.6±4.1	15.7±3.0	**15.6**±3.6	26.2±4.0	27.2±3.2	**25.3**±2.7
GNN-S_1^{Te}	NDCG($\mathcal{V}^R$) ↑	85.9±0.5	85.8±0.3	85.9±0.3	85.8±0.1	85.4±0.3	85.5±0.3	89.8±0.1	89.9±0.1	89.9±0.1
	NDCG($\mathcal{V}_1^R$) ↑	85.9±0.5	85.9±0.3	85.9±0.3	85.9±0.2	85.5±0.3	85.5±0.3	89.8±0.1	89.9±0.1	89.9±0.1
	Δ_{NDCG} ↓	42.6±2.8	**40.7**±3.7	**40.7**±3.7	16.0±3.9	**15.5**±3.0	15.6±3.3	26.1±3.5	27.1±2.4	**24.9**±2.7

Table 5: NDCG values for the outputs of detectors on Yelp datasets with a user degree cutoff at the 10-percentile for defining the group of A. The table displays the mean and standard deviation of NDCG scores for all reviews (NDCG($\mathcal{V}^R$) ↑), for the group with $A = 1$ (NDCG($\mathcal{V}_1^R$) ↑), and the NDCG difference between the two groups (Δ_{NDCG} ↓) across all ten splits. "↑" means the larger, the better; "↓" means the opposite. Our method is denoted by "*". A smaller Δ_{NDCG} indicates a fairer model.

Detector	Metrics(%)	Chi	NYC	Zip
FairGNN [7]	NDCG($\mathcal{V}^R$) ↑	86.2±0.2	85.9±0.1	89.9±0.0
	NDCG($\mathcal{V}_1^R$) ↑	86.4±0.2	86.0±0.1	90.0±0.0
	Δ_{NDCG} ↓	33.8±5.4	22.4±1.9	24.8±1.6
EDITS [8]	NDCG($\mathcal{V}^R$) ↑	84.3±0.2	85.0±0.1	89.2±0.0
	NDCG($\mathcal{V}_1^R$) ↑	84.5±0.2	85.2±0.1	89.4±0.0
	Δ_{NDCG} ↓	37.1±2.0	30.0±1.2	28.9±1.3

Detector	Metrics(%)	GNN(g_θ) w/o	GNN(g_θ) Pre-trained	GNN(g_θ) Joint*	GNN(g_θ) w/o	GNN(g_θ) Pre-trained	GNN(g_θ) Joint*	GNN(g_θ) w/o	GNN(g_θ) Pre-trained	GNN(g_θ) Joint*
GNN	NDCG($\mathcal{V}^R$) ↑	85.4±0.5	84.7±1.3	84.9±1.4	84.8±0.4	84.5±0.3	84.6±0.4	89.7±0.3	89.6±0.6	88.9±0.6
	NDCG($\mathcal{V}_1^R$) ↑	85.7±0.4	85.3±0.8	85.4±0.9	84.8± 0.4	84.6±0.3	84.7±0.4	89.8±0.3	89.6±0.6	89.8±0.4
	Δ_{NDCG} ↓	34.7±1.5	**33.9**±2.5	34.7±1.5	19.7±1.6	19.1±1.6	**18.7**±2.0	25.0±1.8	23.7±1.7	**23.6**±1.7
GNN-S_1^{Tr}*	NDCG($\mathcal{V}^R$) ↑	85.9±0.4	86.1±0.2	86.1±0.2	85.9±0.1	85.9±0.1	85.9±0.1	89.7±0.1	89.9±0.0	89.9±0.0
	NDCG($\mathcal{V}_1^R$) ↑	85.9±0.4	86.2±0.2	86.2±0.2	86.0±0.1	86.0±0.1	86.0±0.1	89.8±0.1	90.0±0.0	90.0±0.0
	Δ_{NDCG} ↓	34.1±4.6	33.7±3.0	**33.2**±2.6	19.1±1.9	16.8±1.5	**16.6**±1.9	25.1±1.6	23.4±1.6	**23.3**±1.4
GNN-S_0^{Te}	NDCG($\mathcal{V}^R$) ↑	85.8±0.5	86.1±0.3	86.1±0.2	85.9±0.1	85.4±0.3	85.4±0.3	89.8±0.1	90.0±0.1	89.9±0.1
	NDCG($\mathcal{V}_1^R$) ↑	85.9±0.5	86.2±0.3	86.2±0.2	86.0±0.1	85.5±0.3	85.5±0.3	89.9±0.1	90.0±0.1	90.0±0.1
	Δ_{NDCG} ↓	33.6±4.4	34.1±3.3	**33.8**±3.0	19.7±2.0	**18.6**±1.5	**18.6**±1.9	25.1±1.6	23.7±1.6	**23.6**±1.4
GNN-S_1^{Te}	NDCG($\mathcal{V}^R$) ↑	85.6±0.5	85.0±1.2	85.0±1.2	85.9±0.1	84.7±0.3	84.7±0.3	89.6±0.1	89.7±0.5	89.5±1.2
	NDCG($\mathcal{V}_1^R$) ↑	85.7±0.6	85.4±0.8	85.5±0.7	86.0±0.1	84.8±0.4	84.8±0.4	89.7±0.1	89.8±0.5	89.2±0.7
	Δ_{NDCG} ↓	**34.0**±4.6	34.3±3.0	34.2±2.8	19.6±1.9	**18.7**±1.7	**18.7**±2.1	25.1±1.6	**23.8**±1.7	24.2±2.8

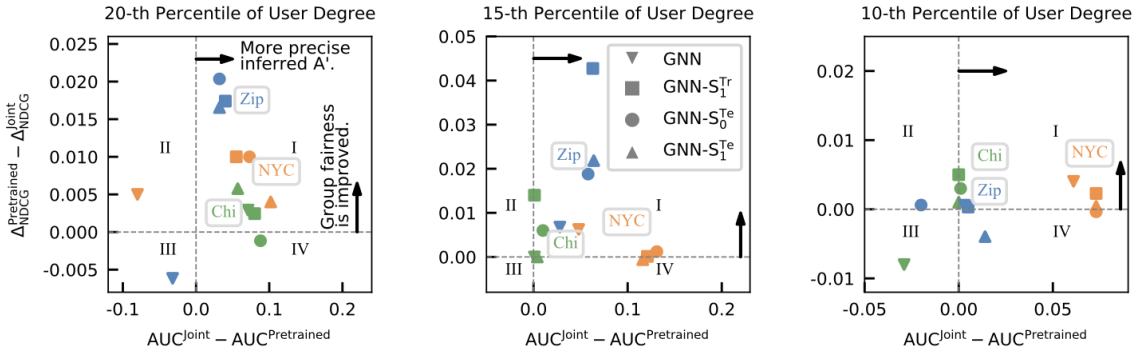


Figure 4: The relationship between the accuracy of predicting A' and group fairness. The x-axis represents the gap in AUC between g_θ's predictions of A' for the Joint and Pre-trained, i.e., $\text{AUC}^{\text{Joint}} - \text{AUC}^{\text{Pretrained}}$. Joint infers a more accurate A' compared to Pretrained if this AUC gap is lager than 0. The y-axis represents the gap in Δ_{NDCG} for spam detector f_W between Pre-trained and Joint, i.e., $\Delta_{\text{NDCG}}^{\text{Pretrained}} - \Delta_{\text{NDCG}}^{\text{Joint}}$. Joint is relatively more fair compared to Pretrained if this Δ_{NDCG} gap is larger than 0. It is evident that the Joint method effectively improves both the AUC of predicted A' and group fairness simultaneously.

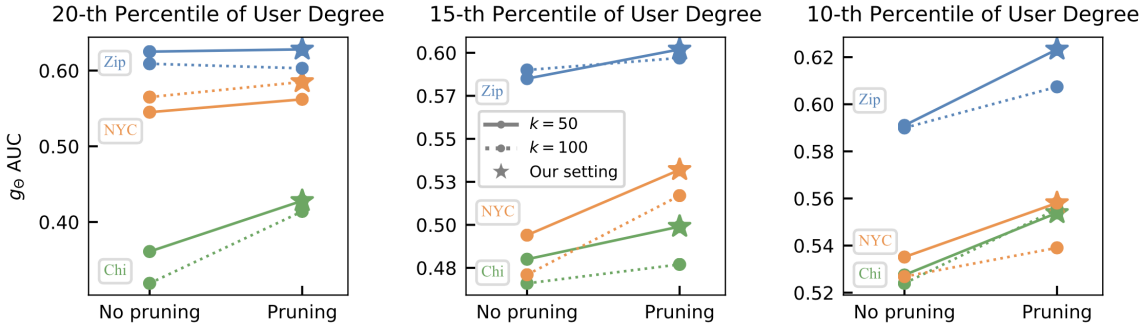
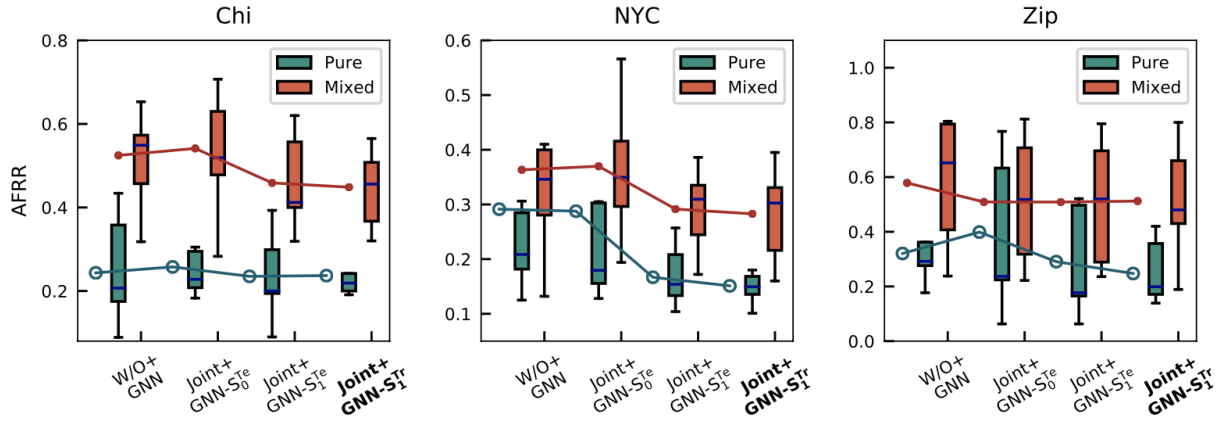
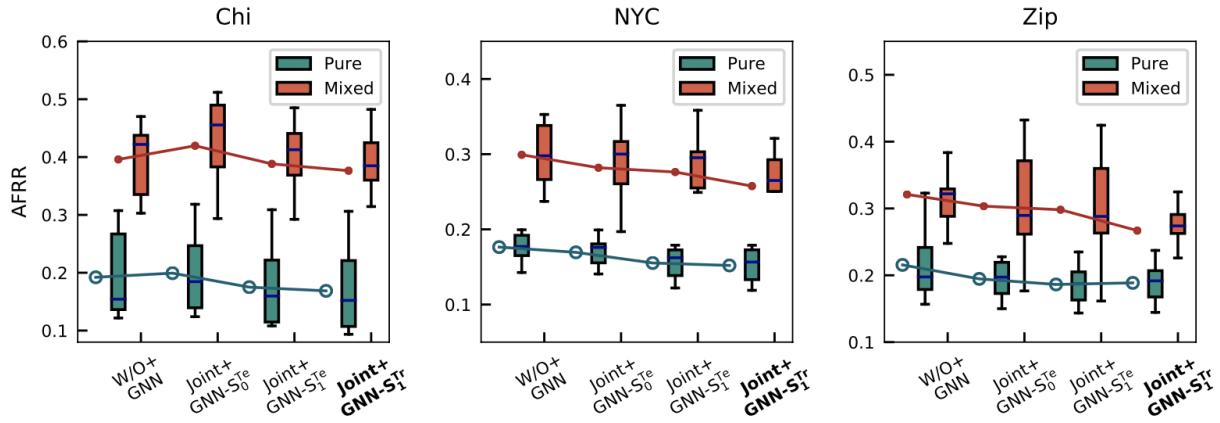


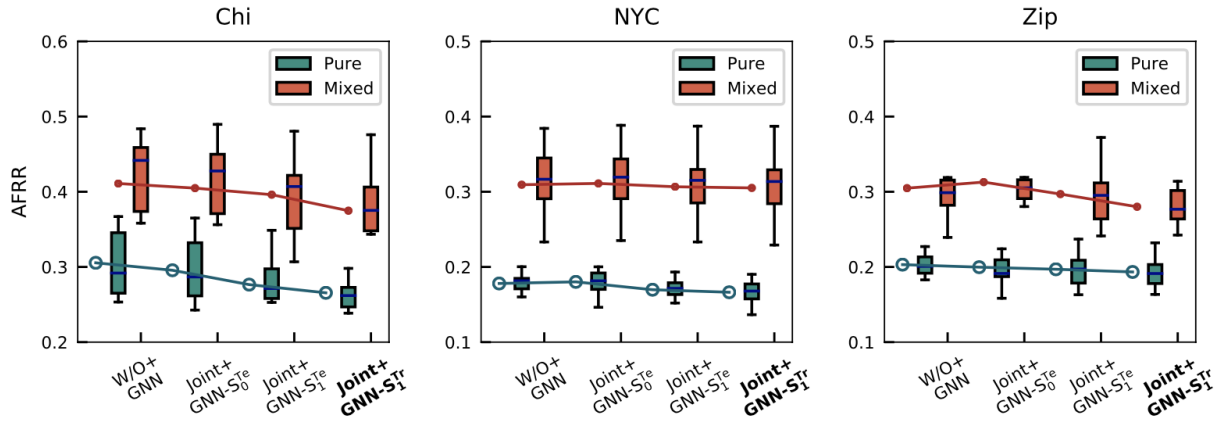
Figure 5: The sensitivity analysis for the replication time k and pruning non-spam edges is presented in this figure. It illustrates the test AUCs ↑ of g_θ on graphs with different values of k, as well as with or without pruning edges. Pruning generally yields better AUCs compared to no pruning.



(a) 20-th Percentile of User Node Degree

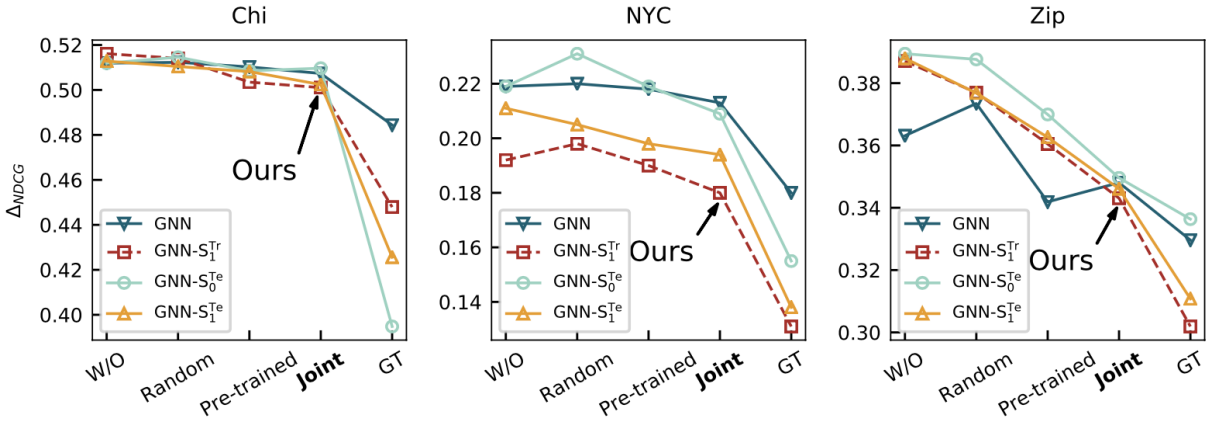


(b) 15-th Percentile of User Node Degree

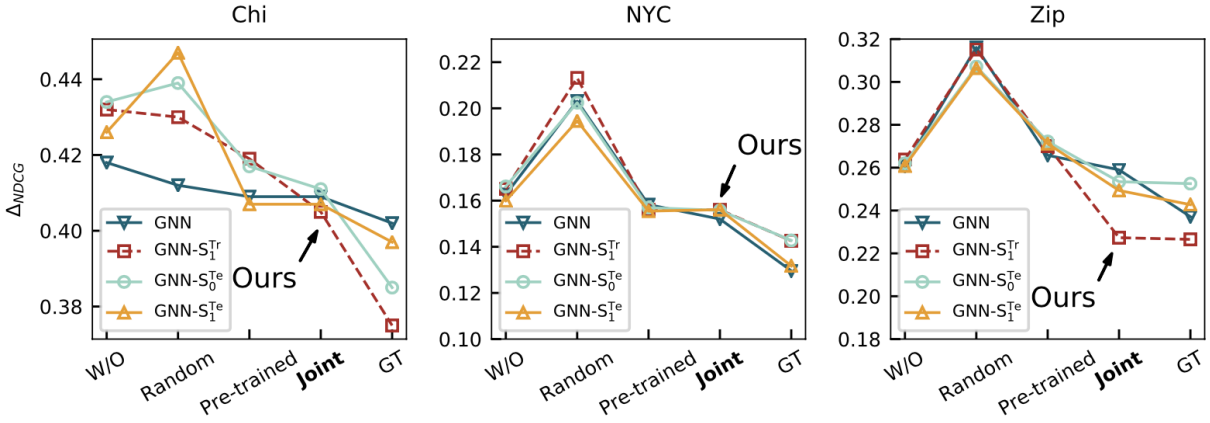


(c) 10-th Percentile of User Node Degree

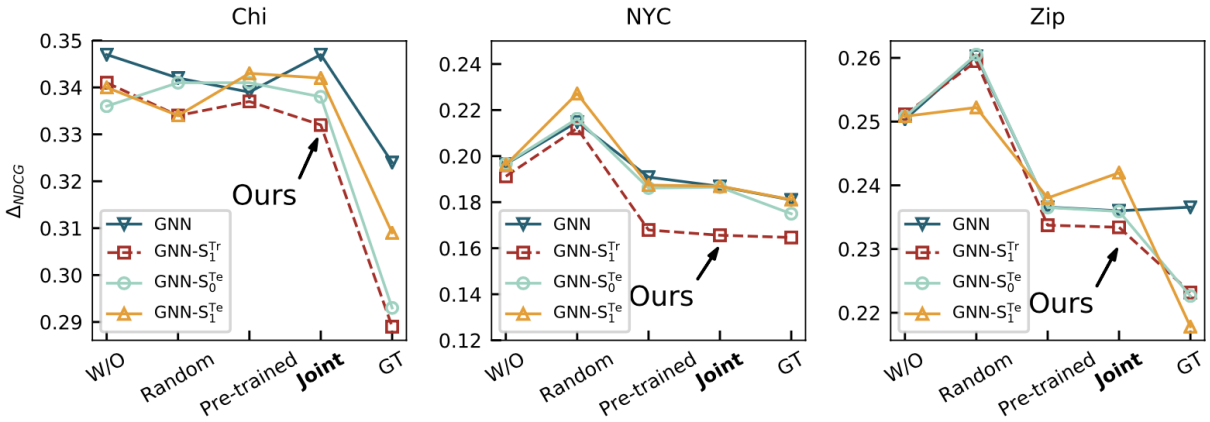
Figure 6: The box plot displays the AFRR values, as defined in Eq. (12), for both test mixed and pure users across all splits of the three datasets under four settings (the proposed method is highlighted in bold). Each box represents a set of AFRRs for all the splits, and the solid lines indicate the mean AFRR for each box. By introducing A' and employing the joint training of f_W and g_θ (referred to as the theJoint method), the AFRR for mixed users decreases as their spam reviews receive higher suspiciousness compared to non-spam reviews.



(a) 20-th Percentile of User Node Degree



(b) 15-th Percentile of User Node Degree



(c) 10-th Percentile of User Node Degree

Figure 7: This figure presents the test $\Delta_{\text{NDCG}} \downarrow$ for four detectors: GNN, GNN-S_1^{Tr} (shown as a dashed line, representing our method), GNN-S_0^{Te}, and GNN-S_1^{Te}. These detectors utilize A' obtained from five different methods. The amount of noise in A' gradually decreases from left to right, corresponding to the methods: w/o (without A'), Random (randomly assigning $A' = 1/0$), Pre-trained (output of a pre-trained g_θ), Joint (output of jointly trained g_θ), and GT (ground truth of A'). When the inferred A' is accurate, it leads to a decrease in Δ_{NDCG}. Our method shows the smallest Δ_{NDCG} in addition to the NDCG given by the ideal GT method.